Summary of Progress

In this report, we will focus on the results included in the Ph.D. dissertation of Dr. Fu-Quan Wang, who was supported by the grant as a Research Assistant from January 1989 through December 1992. Dr. Wang completed his dissertation and received his Ph.D. degree in December 1992. A copy of the dissertation is included as an Appendix to this report. One journal paper has been accepted for publication based on this research [1], another has been submitted for publication [2], and three more are in preparation for submission [3-5]. In addition, several conference presentations have resulted from this work [6-10]. The following sections contain a brief summary of the important aspects of this dissertation.

1) Erasurefree Sequential Decoding of Trellis Codes

The publication of Ungerboeck's seminal paper [1] on trellis coded modulation stimulated wide interest in the construction of good trellis codes. However, very few papers have addressed the decoding problem. Most researchers assume that the Viterbi Algorithm (VA) is used for decoding and trellis codes are then constructed by hand or by computer search to maximize the minimum free Euclidean distance and/or minimize the number of nearest neighbors. However, since both the hardware complexity and the computational effort of the VA increase exponentially with the constraint length, ν, it is not practical to implement the VA for large ν and its performance is limited to moderate bit error rates (BER's). To achieve better performance requires the use of larger constraint lengths and suboptimum decoding.

It is well known that the computational effort and the hardware complexity of Sequential Decoding (SD) algorithms are essentially independent of the constraint length ν, so large ν can be used and arbitrarily small error probability can be obtained with reasonable complexity and high decoding speed. Unfortunately, though, the computational effort of SD is a random variable with a Pareto distribution. Although the undetected error probability can be made arbitrarily small, some data cannot be completely decoded and the probability of incomplete decoding (erasure) is usually on the order of 10^{-2} to 10^{-3}. Thus, the performance of SD is limited in the case where a feedback channel is not available. However, if the drawback of erasures can be overcome, SD may be a good alternative to the VA even if a feedback channel is not available.

The two most popular sequential decoding algorithms are the Fano Algorithm (FA) and the Stack Algorithm (SA). The SA always extends the path with the best metric until it reaches the terminal node. All previously extended paths must be stored during the process of decoding. The storage of extended paths forms an ordered stack which may overflow if it is finite. On the other hand, because it does not require any storage, the FA does not have a stack overflow problem. In order to insure extending the path with the best metric (the top path), the SA requires a large effort to continually re-order the stack. This problem can be partially solved by using the stack bucket algorithm, but the FA still decodes faster than the SA for rates below the channel cut-off rate. Thus the FA is preferred in most practical implementations, and we have chosen to focus on a Fano-type algorithm.

The cut-off rate for two-dimensional modulation channels with equiprobable signaling has been computed. (The cut-off rate of a channel is the maximum rate that is practically achievable using sequential decoding.) For two dimensional modulation channels, we find that almost all the potential coding gain of trellis coding with sequential decoding is achieved by doubling the number of channel signals, as is the case with channel capacity.

Some practical considerations in the application of sequential decoding to trellis codes were also investigated. The Fano metric was derived and several quantization schemes were studied via simulation for PSK constellations.A simple method to increase the distance of trellis codes in the tail was developed and the influence of the tail on performance was studied. Simulation results for large constraint length trellis codes using sequential decoding show that performance improves with increasing constraint length and significant coding gains over Viterbi decoding can be achieved.

A general erasurefree sequential decoding scheme called the Buffer Looking Algorithm (BLA) has been proposed and the resynchronization problem of sequential decoding has been addressed. The BLA in a Block Decoding mode (BLA-BD) guarantees resynchronization at the beginning of each block but suffers some rate loss, i.e., has a lower spectral efficiency. The performance of the BLA-BD has been analyzed and simulations have been performed. They show that the BLA-BD with a constraint length $\nu = 13$ code, a block length of 256 symbols, and a speed factor of 6 can achieve about 1.1 dB coding gain at a BER of 10^{-5} over the Vitgerbi Algorithm (VA) with a constraint length $\nu = 6$ (64 state) code. The VA with a 64 state code requires 64 computations to decode one branch, which is substantially larger than the maximum average number of computations (speed factor) of 6 per branch for the BLA-BD. The BLA-BD can achieve the channel cut-off rate bound and a full 1.4 dB coding gain over the 64 state VA when larger constraint length codes are used. If the rate loss is taken into account, more than 1 dB of coding gain over the VA can still be achieved.

A general resynchronization scheme has been presented for continuous sequential decoding. It was shown that sequential decoding using this scheme has a high probability of resynchronizing successfully. This solves the rate loss problem resulting from the block decoding approach. (Using this resynchronization scheme in the BLA-BD with a large block length may provide the best way to achieve a flexible trade-off between rate loss and error performance in sequential decoding.) The performance of the BLA in a Continuous Decoding mode (BLA-CD) using the resynchronization scheme was studied via simulations. They show that the BLA-CD performs about as well as the BLA-BD at a BER of 10^{-5} and has a slightly larger spectral efficiency.

2) Probabilistic Construction of Trellis Codes

Although many tellis codes have been constructed, few of them are intended for use with sequential decoding. Porath and Aulin [12] proposed non-exhaustive search code construction algorithms for finding good long systematic feedback trellis codes. Their algorithms are a generalization of the Lin and Lyne algorithm [13]. (The Lin and Lyne algorithm has also been used to construct feedforward trellis codes for use with sequential decoding in [1].) This type of algorithm guarantees that codes with good column distance growth are found and thus is a good choice for constructing convolutional or trellis codes for use with sequential decoding. However, it is the code free distance that determines the error performance and the Lin and Lyne algorithm cannot guarantee that codes with large free distance are found. Furthermore, it is very difficult to evaluate the free distnace of codes with large constraint length. This poses a problem for the selection of good codes using any conventional code construction algorithm. Thus, we investigated a probabilistic approach to constructing good large constraint length trellis codes for use with sequential decoding.

Simulation results for trellis codes show that many randomly chosen codes perform very well. (This is consistant with what is expected from the random coding bound.) Two probabilistic construction algorithms were proposed to randomly construct large constraint length trellis codes for use with sequential decoding and trellis codes for 8-PSK modulation and 16-QAM modulation with constraint lengths up to $v = 20$ were obtained. The new short constraint length codes were compared to the best known codes with short constraint lengths. The results showed that the new codes perform almost as well as the best known codes at a BER of 10^{-5}. Simulations were then used to show that the cut-off rate bound can be achieved using the new large constraint length trellis codes with sequential decoding at BER's of 10^{-5} to 10^{-6}. Up to 6.6 dB real coding gains over an uncoded system and up to 2.0 dB real coding gains over 64-state trellis codes using Viterbi decoding can be achieved when the new codes are used with sequential decoding.

3) Construction of Robustly Good Trellis Codes

The free distance has been used as the main criterion in the construction of trellis codes for use with the VA. Since the computational effort for sequential decoding is a random variable, parameters relating to the computational distribution of trellis codes should also be taken into account in the selection of trellis codes for use with sequential decoding. We have investigated the relationship between the computational effort of sequential decoding and the column distance function of trellis codes and determined the best design criteria for trellis codes with sequential decoding.

The influence of the column distance function and the distance profile of trellis codes on the computational effort of sequential decoding was studied by analysis and simulation. We found that codes with a rapidly growing column distance function result in better computational performance and that the initial portion of the column distance function (i.e., the distance profile) plays a more important role than its latter part.

Trellis codes with Optimum Distance Profiles (ODP) and Optimum Free Distances (OFD) for 8-PSK and 16-QAM modulation with constraint lengths up to 15 have been constructed for use with sequential decoding. Although they provide a better trade-off between the free distance and the distance profile than the best known trellis codes constructed for the VA, neither the ODP nor the OFD trellis codes provide the best trade-off, i.e., the distance profiles of some OFD trellis codes are much worse than the ODP codes, and the free distances of some ODP trellis codes are much worse than the OFD codes. This is quite different from the case with convolutional codes, where the best free distance codes also have good distance profiles.

Thus, we have constructed trellis codes which are neither optimum free distance nor optimum distance profile. We call the new codes Robustly Good Codes (RGC). Given that a robustly good trellis code of constraint length ν has been found, the approach used to find a constraint length $\nu + 1$ robustly good trellis code is to find the code that improves the free distance or the distance profile of the constraint length ν code, with priority given to improving the free distance. In other words, we try to find a longer code which has a free distance or a distance profile superior to or identical to the shorter one. The new codes achieve nearly the same free distances as the OFD codes and nearly the same distance profiles as the ODP codes. Simulation results show that the new codes outperform the best known trellis codes when sequential decoding is used.

4) On the Separability of Shaping and Coding

In a coded modulation system, a shaping gain can be achieved by using either higher dimensional spherical constellations or appropriately designed shaping codes. However, it has been recognized that it is advantageous to pursue shaping gain directly via a shaping code rather then indirectly via shaping a higher dimensional constellation. Existing schemes that employ shaping and coding utilize one or more normal codes and a shaping code separately. Forney [1[illegible]] asserts that shaping and coding are separable and their gains additive at high data rates (spectral efficiencies). However, Pottie and Calderbank [15] recently argued that shaping and coding may not be separable in the limit of large code complexity. We have investigated the separability of shaping and coding in a coded/shaped system operating at practical spectral efficiencies (≤ 8 bits per two dimensional signal). We have shown in this case that coding gain and shaping gain in a separated system are not additive. We have also shown that a separated coded/shaped system cannot achieve Shannon's bound on performance. A new cascade structure for combined coding and shaping has been proposed. Although it may be possible to achieve Shannon's bound using this structure, the design of a shaping scheme for this structure remains an open problem.

References

[1] S. S. Malladi, F. Q. Wang, D. J. Costello, Jr., and H. C. Ferreira, "Construction of Trellis Codes with a Good Distance Profile", *IEEE Transactions on Communications*, to appear.

[2] F. Q. Wang and D. J. Costello, Jr., "Erasurefree Sequential Decoding of Trellis Codes", *IEEE Transactions on Information Theory*, submitted for publication, February 1993.

[3] F. Q. Wang and D. J. Costello, Jr., "Construction of Trellis Codes for Sequential Decod-

ing", *IEEE Transactions on Communications*, to be submitted.

[4] F. Q. Wang and D. J. Costello, Jr., "Construction of Robustly Good Trellis Codes", *IEEE Transactions on Information Theory*, to be submitted.

[5] F. Q. Wang and D. J. Costello, Jr., "On the Separability of Shaping and Coding", *IEEE Transactions on Information Theory*, to be submitted.

[6] F.Q. Wang and D. J. Costello, Jr., "Erasurefree Sequential Decoding and Its Application to Trellis Codes", Abstracts of Papers, *IEEE International Symposium on Information Theory*, p. 66, San Diego, CA, January 1990.

[7] F. Q. Wang and D. J. Costello, Jr., "A Hybrid M-Algorithm/Sequential Decoder for Convolutional and Trellis Codes", *Proceedings of the International Symposium on Information Theory and Its Applications*, pp. 67-69, Honolulu, HI, November 1990.

[8] F. Q. Wang and D. J. Costello, Jr., "Probabilistic Construction of Trellis Codes", *Proceedings of the IEEE International Symposium on Information Theory*, p. 200, Budapest, Hungary, June 1991.

[9] F. Q. Wang and D. J. Costello, Jr., "On the Design Criteria for Trellis Codes with Sequential Decoding", *Proceedings of the IEEE Internation Symposium on Information Theory*, p. 415, San Antonio, Texas, January 1993.

[10] F. Q. Wang and D. J. Costello, Jr., "A General Resynchronization Scheme for Sequential Decoding", *Proceedings of the Conference on Information Sciences and Systems*, The Johns Hopkins University, Baltimore, Maryland, March 1993, to appear.

[11] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", *IEEE Transactions on Information Theory*, IT-28, pp. 55-67, January 1982.

[12] J. Porath and T. Aulin, "Algorithmic Construction of Trellis Codes", *IEEE Transactions on Communications*, submitted for publication, November 1990.

[13] S. Lin and H. Lyne, "Some Results on Binary Convolutional Code Generators", *IEEE Transactions on Information Theory*, IT-13, pp. 134-139, January 1967.

[14] G. D. Forney, Jr., "Trellis Shaping", *IEEE Transactions on Information Theory*, IT-38, pp. 281-300, March 1992.

[15] G. J. Pottie and A. R. Calderbank, "Asymptotic Upper Bounds on the Minimum Distance of Trellis Codes", *Proceedings of the IEEE International Symposium on Information Theory*, p. 148, Budapest, Hungary, June 1991.

Appendix

Efficient Sequential Decoding of Trellis Codes

EFFICIENT SEQUENTIAL DECODING OF TRELLIS CODES

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Fu-Quan Wang, BSEE, MSEE

Daniel J. Costello Jr.

Daniel J. Costello, Jr., Director

Department of Electrical Engineering
Notre Dame, Indiana
December, 1992

EFFICIENT SEQUENTIAL DECODING OF TRELLIS CODES

Abstract

by

Fu-Quan Wang

The application of sequential decoding to trellis codes is studied. It is shown that sequential decoding is a good alternative to Viterbi decoding and the results conform closely to experience with convolutional codes.

An erasurefree sequential decoding algorithm is introduced. Analysis and simulation show that significant coding gains over Viterbi decoding can be achieved with much less computational effort using the new algorithm.

Trellis codes for 8-PSK and 16-QAM modulation with optimum distance profile and optimum free distance are constructed. The design criteria for trellis codes with sequential decoding are examined. A new code construction algorithm is proposed to construct robustly good trellis codes for use with sequential decoding. Trellis codes with asymptotic coding gains up to 6.66 dB are obtained.

Probabilistic construction algorithms are investigated for constructing good large constraint length trellis codes that can achieve the channel cut-off rate at a bit error rate of $10^{-5} - 10^{-6}$. Codes for 8-PSK and 16-QAM modulations with constraint lengths ν up to 20 are obtained. Simulation results show that the codes can achieve the cut off rate bound at a bit error rate of $10^{-5} - 10^{-6}$ which correspond to $5.3 - 6.6$ dB real coding gains over uncoded systems.

Relationship between shaping and coding is studied and the separability of shaping and coding in a coded/shaped modulation system is examined.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

If I looked farther, it is because I am standing on the shoulder of my advisor, Professor Daniel J. Costello, Jr. I would like to thank him for his advice, comments, suggestions, and stimulating discussions during my research which leads to this dissertation.

I would like to thank Lance C. Perez for his assistance and helpful discussions during this study.

I would like to thank Professors Mark A. Herro, Ken D. Sauer, and Robert L. Stevenson for serving on my dissertation committee.

I am grateful to Professor Chang-nian Cai and many other faculty members at the Beijing University of Post and Telecommunications for their teachings, support, and encouragement at the earlier stage of my academic career, without which I would not have been where I am.

Last but not least, I would like to thank the National Aeronautics and Space Administration (grant number NAG 5–557) and the National Science Foundation (grant number NCR 89–03429) for their financial support, which makes this investigation possible.

1

INTRODUCTION

It has been predicted that wireline as well as wireless communications will be fully digital by the end of this century[75]. Digital communications provides excellent reproduction of the source signals with the greatest efficiency of transmission bandwidth and power by using source and channel coding. Source coding reduces the transmission rate for a given degree of fidelity [1, 30]. Channel coding can reduce the Signal-to-Noise Ratio (SNR) and bandwidth requirements for a given degree of reliability [9, 46, 70]. In this dissertation, we will study the decoding and construction of a class of channel codes.

The reliability of a digital communication system is usually measured by the bit error rate P_b which is defined as the total number of error bits over the total number of transmitted information bits. The power efficiency is reflected by the SNR per bit and the bandwidth efficiency is measured by the number of bits that can be transmitted by a two dimensional signal. Many efforts[9, 46] have been undertaken to achieve large power efficiency using coding for power limited channels at the expense of bandwidth. However, the application of these codes to bandwidth limited channels is not successful. It is the work of Ungerboeck[70] that showed how both power and bandwidth efficiencies can be achieved for bandwidth limited channels. The Ungerboeck codes are usually called Trellis Coded Modulation (TCM) which is a subclass of the so called trellis codes.

Usually the Viterbi decoding algorithm[73] , which is optimum in the sense of being maximum likelihood, is used to decode trellis codes. One drawback of the Viterbi algorithm is the exponential growth of its computational effort with the code constraint length. To achieve larger coding gains, alternative decoding algorithms must be explored. In this dissertation, we investigate the application of sequential decoding to trellis codes and the construction of trellis codes for use with sequential decoding.

1.1 Digital Communication Systems

A typical digital communication system is depicted in Figure 1.1. The source could either be a person or a machine that generates a sequence of messages to be communicated to the receiving terminal. The output message of the source could either

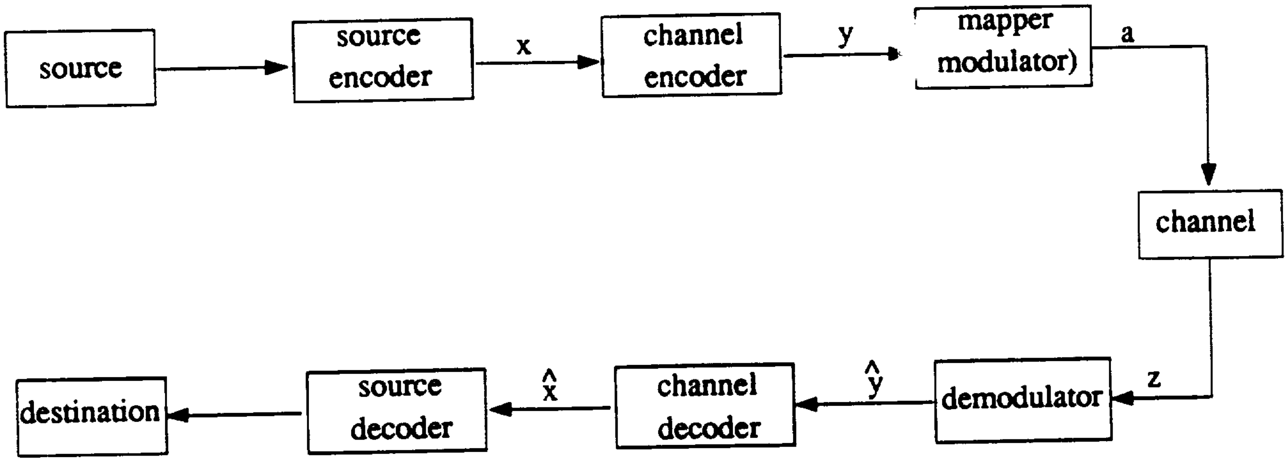


Figure 1.1: Block diagram of a digital communication system

be continuous signals or a sequence of discrete symbols. The amount of information generated by the source can be measured by the entropy of the signal set for a discrete source or the rate distortion function for a continuous source [28].

The source encoder transforms the output of source into a sequence of binary digits (bits) called information sequence $\mathbf{x}$. The encoder is designed such that the minimum number of bits is required to represent the source output for a discrete source. This can be accomplished using entropy coding algorithms (encoder) for a discrete source. For a continuous source, the encoder is designed such that the minimum number of bits required to represent the source output with a predetermined distortion (fidelity) is achieved. Theory and practical techniques for such transformations have been well developed [1, 30]. Generally speaking, the source encoder tries to represent the source output as economically as possible.

The channel encoder transforms the information sequence $\mathbf{x}$ into a code sequence $\mathbf{y}$. The code sequence $\mathbf{y}$ is then mapped into a sequence of modulated signals that are suitable for transmission in physical channels.

To transmit n bits/T (T is the modulation time period), 2^n distinctive functions $\{s_i(t), i = 0, 1, \cdots, 2^n - 1\}$, which are suitable for transmission in physical channels, are needed. A vector of n bits selects one of the functions $\{s_i(t)\}$ at modulation time lT. Using Gram-Schmidt orthogonalization procedure[89], $\{s_i(t)\}$ can be expressed as N-dimensional vectors $\{a^i, i = 0, 1, \cdots, 2^n - 1\}$ $(N \leq 2^n)$. The receiver error probability is determined by $\{a^i\}$, i.e., only the vectors $\{a^i\}$ are important. $\{a^i\}$ is called the constellation of the modulation scheme, which can be displayed in a R^N Euclidean space. In digital communication systems, two dimensional signals are usually used. Some typical constellations are shown in Figure 1.2.

Traditionally, the channel encoder is designed such that the minimum Hamming distance among the code sequences $\mathbf{y}$'s, which is called the free Hamming distance of

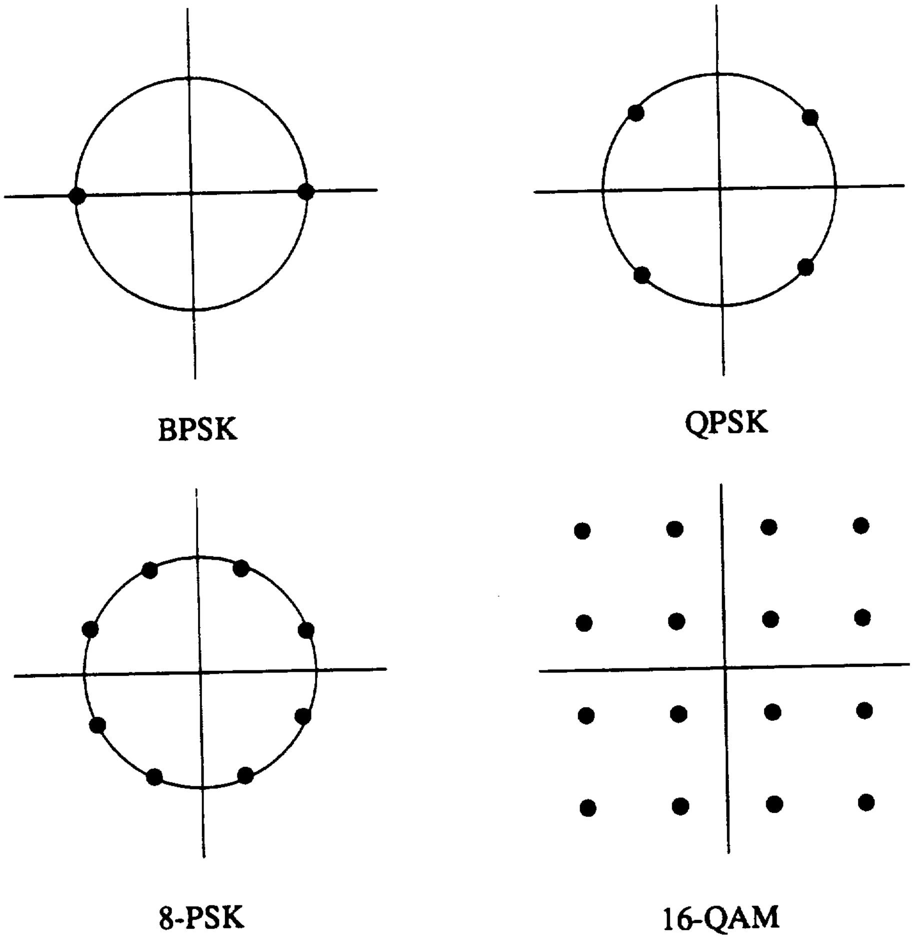


Figure 1.2: Typical modulation constellations

the code for convolutional codes, are maximized. The minimum Euclidean distance among the sequence **a** is equivalent to the minimum Hamming distance among the code sequences **y** if and only if BPSK or (Gray mapped) QPSK modulation is used. We define the number of information bits that can be transmitted per modulation time period as the spectral efficiency. For uncoded BPSK and QPSK, spectral efficiencies of 1 bit/T and 2 bits/T can be achieved, respectively. However, when coding is used, some redundant bits are introduced in $\mathbf{y}$. Thus it is impossible for the digital communication systems using traditional codes to achieve a spectral efficiency of 2 bits/T or larger. When the channel is bandlimited, we wish to achieve better spectral (bandwidth) efficiency. But, for other modulation constellations, large Hamming distance among the code sequences $\mathbf{y}$ does not necessarilly result in large Euclidean distance among the modulated sequence $\mathbf{a}$. Thus, it is desirable to optimize the Euclidean distance of the code sequence directly. The design of codes that have large Euclidean distance and the application of sequential decoding to these codes are the subject of this dissertation.

The modulated signals or the output of the trellis encoder are then transmitted through a physical channel and is corrupted by the noise. The demodulator acts as an optimum receiver which usually includes a matched filter or a correlation detector followed by a sampling switch. The output of the matched filter or a correlation detector is sampled at time lT. The resulted signal z_l is a discrete symbol. The optimum receiver (demodulator) is designed such that the signal to noise ratio is maximized.

The channel decoder transforms a sequence of real numbers $\mathbf{z}$ or its quantized version $\hat{\mathbf{y}}$ into a sequence of binary digits $\hat{\mathbf{x}}$ which is the estimate of $\mathbf{x}$. The number of the positions where $\hat{\mathbf{x}}$ and $\mathbf{x}$ differ are the number of bit errors made by the communication system. The number of errors divided by the total number of bits for the sequence $\mathbf{x}$ is defined as the Bit Error Rate (BER) of the system. BER is a very important measure of the quality of a digital communication system. Obviously, an optimum decoder is the one that minimizes the BER. BER is related to the free Euclidean distance of the code and the channel signal to noise ratio. This will be discussed in detail later in the dissertation.

The source decoder transforms the sequence $\hat{x}$ into an estimate of the source output and deliver this estimate to the destination.

1.2 The Capacity and Cut-off Rate of Two Dimensional Modulation Channels

The trellis coded communication system model considered in this dissertation is depicted in Figure 1.3. The information sequence x from the source is divided into

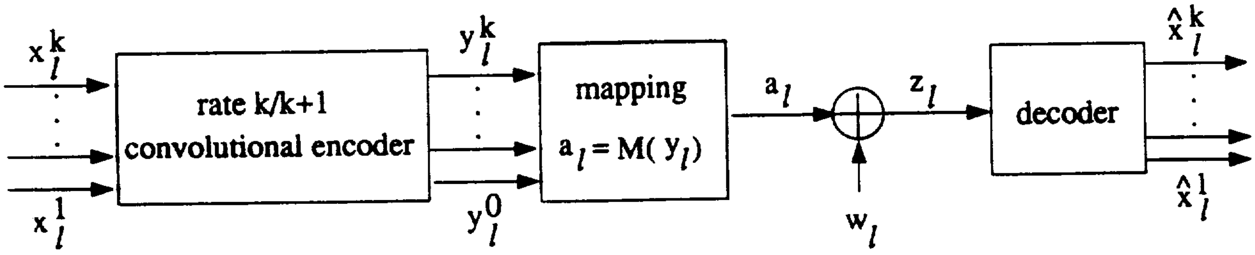


Figure 1.3: Block diagram of a trellis coded digital communication system

k subsequences and fed into a rate $k/k+1$ convolutional encoder whose outputs are then mapped into signals suitable for transmission over a physical channel. The combination of the convolutional encoder and the signal mapper is called the trellis encoder.

Suppose that the channel input signal a_l at modulation time l is taken from a collection of signals $\{a^i, i = 0, 1, \cdots, \text{K-1}\}$ with probability $p_i = P\{a_l = a^i\}$ (i=0,1, $\cdots$, K-1), where a^i is represented as a point in a two dimensional constellation (such as PSK or QAM) and K is the number of points in the constellation. Actually, the

a^i, for $i = 0, 1, \cdots$, K-1, are K distinct continuous time functions of duration T. They are usually sine wave pulses with different amplitudes and/or phases. However, we can represent them as two dimensional points in a signal constellation[89] and thus regard them as discrete symbols. The average signal energy is then given by

$$E_S = \sum_{i=0}^{K-1} p_i ||a^i||, \tag{1.1}$$

where $||x||$ denotes the energy of signal x. If $p_i = 1/K$ for (i=0,1, $\cdots$, K-1), we say that equiprobable signaling is used. Otherwise, we say that nonequiprobable signaling is used.

The modulated signal sequence is then transmitted through a physical channel and is corrupted by noise. Let w_l be a bandlimited Additive White Gaussian Noise (AWGN) sample at time l with zero mean and variance σ^2 per dimension. Assume that distortionless transmission, perfect timing, and carrier-phase synchronization are available. Then, the channel output (demodulated signal) at the receiver at time l is

$$z_l = a_l + w_l \tag{1.2}$$

and has probability density:

$$p\{z_l/a_l = a^i\} = \frac{1}{2\pi\sigma^2} exp\{\frac{-|z_l - a^i|^2}{2\sigma^2}\}. \tag{1.3}$$

The average SNR per symbol is defined as

$$SNR = E_S/2\sigma^2. \tag{1.4}$$

It is well known[89] that there are two parameters that determine the fundamental performance limits for digital communications: the channel capacity C and the channel cut-off rate R_0. Shannon[68] proved that reliable communication can be achieved through coding when the transmission rate is less than the channel capacity

C. The channel capacity for a bandlimited AWGN channel with ideal signaling and an average input energy constraint E_S is given by[68]

$$C = \log_2\left(1 + \frac{E_S}{N_0}\right) \tag{1.5}$$

in bits per signal duration T (bit/T). But C may only be achieved with Gaussian distributed input signals.

The channel capacity for a discrete-input continuous-output bandlimited AWGN channel with equiprobable signaling can be derived as follows. Assume that the input signals are independent random variables and the channel is memoryless. Then, extension of the formula for the capacity of a discrete memoryless channel [9] to the case of continuous-output yields

$$C^* = \max_{p_0,\cdots,p_{K-1}} \sum_{i=0}^{K-1} p_i \int_{-\infty}^{+\infty} p(z_l/a_l = a^i) \log_2\left\{\frac{p(z_l/a_l = a^i)}{\sum_{j=0}^{K-1} p_j p(z_l/a_l = a^j)}\right\} dz_l \quad bit/T. \tag{1.6}$$

For an equiprobable signaling system, we have $p_i = 1/K$ $(i = 0, 1, \cdots, K-1)$, so the maximization in (1.6) can be omited. By doing some further calculations, we obtain the channel capacity[70]:

$$C^* = \log_2 K - \frac{1}{K}\sum_{k=0}^{K-1} E_z\left\{\log_2 \sum_{i=0}^{K-1} exp\left[-\frac{|z-a^i|^2 - |z-a^k|^2}{2\sigma^2}\right]\right\}. \tag{1.7}$$

where E_z denotes the expectation of z. C^* can be evaluated by Monte Carlo techniques for a given Signal to Noise Ratio (SNR). From (1.7), we see that C^* is a function of the constellation points $\{a^i\}$. Thus, the SNR's required to achieve the same spectral efficiency (number of information bits/T) will differ for different constellations. On the other hand, (1.5) shows that C is only a function of SNR.

From (1.7), we may infer that some constellations are more power efficient than some other constellations. For two dimensional constellations, QAM or its more circular variations are most efficient. However, even in these cases, more energy is required than that promised by (1.5) to achieve the same spectral efficiency when conventional coding (equiprobable signaling) is applied. For example, Table 1.1 shows the required SNR's to achieve the same spectral efficiencies of $C = 2, \cdots, 7$ for ideal (nonequiprobable signaling) and $C^* = 2, \cdots, 7$ for QAM modulation with equiprobable signaling. It is noted that the required SNR for C^* is significantly larger than the required SNR for the same spectral efficiency of C even if a very large constellation is used. The difference in the SNR's needed to achieve the same C^* as C can be viewed as the maximum possible shaping gain with respect to the channel capacity.

Table 1.1: The required SNR to achieve C and C^*

C or C* system SNR	2	3	4	5	6	7
256-QAM	5.1	9.0	12.6	16.0	19.2	22.5
128-QAM	5.1	9.0	12.6	16.0	19.3	–
64-QAM	5.1	9.1	12.7	16.2	–	–
32-QAM	5.1	9.1	12.8	–	–	–
16-QAM	5.2	9.3	–	–	–	–
8-QAM	5.4	–	–	–	–	–
Ideal	4.8	8.5	11.8	14.9	18.0	21.0

A shaping gain can be achieved by using either higher dimensional spherical constellations[5, 22, 27] or appropriately designed shaping codes [3, 4, 24, 48]. However, it has been recognized [4, 24] that it is advantageous to pursue shaping gain directly via a shaping code rather than indirectly via shaping a higher dimensional constellation. The objective of shaping coding is to minimize the average signal power by achieving a nonuniform Gaussian-like input distribution on an expanded constellation. A more conventional shaping gain definition is given by Forney and Wei[27]. In their definition, the shaping gain of an N-dimensional region $\mathbf{R}$ is the reduction in average power (per two dimensions) required by a constellation bounded by $\mathbf{R}$ compared to that which could be required by a constellation bounded by an N-cube of the same volume $V(\mathbf{R})$, i.e.,

$$\gamma_s(\mathbf{R}) = [12G(\mathbf{R})]^{-1}, \tag{1.8}$$

where

$$G(\mathbf{R}) = \frac{\int_{\mathbf{R}} ||r||^2 dV}{NV(\mathbf{R})^{1+2/N}}, \tag{1.9}$$

is the normalized second moment of region $\mathbf{R}$. It has been shown that the shaping gain asymptotically approaches $\pi e/6$ (1.53 dB) as N $\rightarrow \infty$ [26, 27]. This limit is called the ultimate shaping gain.

Channel capacity may only be achieved with infinite coding complexity. Channel cut-off rate R_0 is the maximum rate at which the average number of computations for sequential decoding is bounded. Thus, R_0 is regarded as the maximum rate for which reliable communication can be achieved with reasonable complexity by many authors [52, 90]. The cut-off rate for a bandlimited AWGN channel with ideal signaling and an average input energy constraint E_S is given by[28]

$$R_0 = (\log_2 e)\left[1 + \frac{E_s}{2N_0} - \sqrt{1 + \left(\frac{E_s}{2N_0}\right)^2}\,\right] + \log_2\left[\frac{1}{2}\left(1 + \sqrt{1 + \left(\frac{E_s}{2N_0}\right)^2}\,\right)\right] \quad bit/T. \tag{1.10}$$

But R_0 may only be achieved with Gaussian distributed input signals.

The cut-off rate for a discrete-input continuous-output bandlimited AWGN channel with equiprobable signaling can be derived using a similar approach as the capacity [70]. Extension of the formula for the cut-off rate of a discrete memoryless channel [9] to the case of continuous-valued outputs yields

$$R_0^* = -\log_2\left\{\int_{-\infty}^{+\infty}\left[\frac{1}{K}\sum_{k=0}^{K-1}\sqrt{p(z/a^k)}\right]^2 dz\right\} \quad bit/T. \tag{1.11}$$

Substituting (1.3) into (1.11) and doing some further calculations, we obtain the cut-off rate R_0^* as

$$R_0^* = 2\log_2 K - \log_2\left\{\sum_{i=0}^{K-1}\sum_{j=0}^{K-1} exp\left[-\frac{({a_x}^i - {a_x}^j)^2 + ({a_y}^i - {a_y}^j)^2}{8\sigma^2}\right]\right\}, \tag{1.12}$$

where ${a_x}^i$ and ${a_y}^i$ are the inphase and quadrature components of the two dimensional signal a^i.

In Figure 1.4, R_0^* is plotted as a function of SNR for some two dimensional constellations. R_0 is also plotted in Figure 1.4. Observe that a larger SNR is needed to achieve the same R_0^* as R_0. The difference in the SNR's is caused by the different input signal distributions just as in the case of channel capacities C^* and C. R_0 is only a function of the SNR, whereas R_0^* depends on the specific signal constellation. For a given constellation, R_0^* can be achieved with proper coding. The difference in the SNR's needed to achieve the same R_0^* as R_0 can be viewed as the maximum possible shaping gain with respect to the cut-off rate. Table 1.2 shows the required SNR's to achieve the same spectral efficiencies of $R_0 = 2, \cdots, 7$ for ideal (nonequiprobable

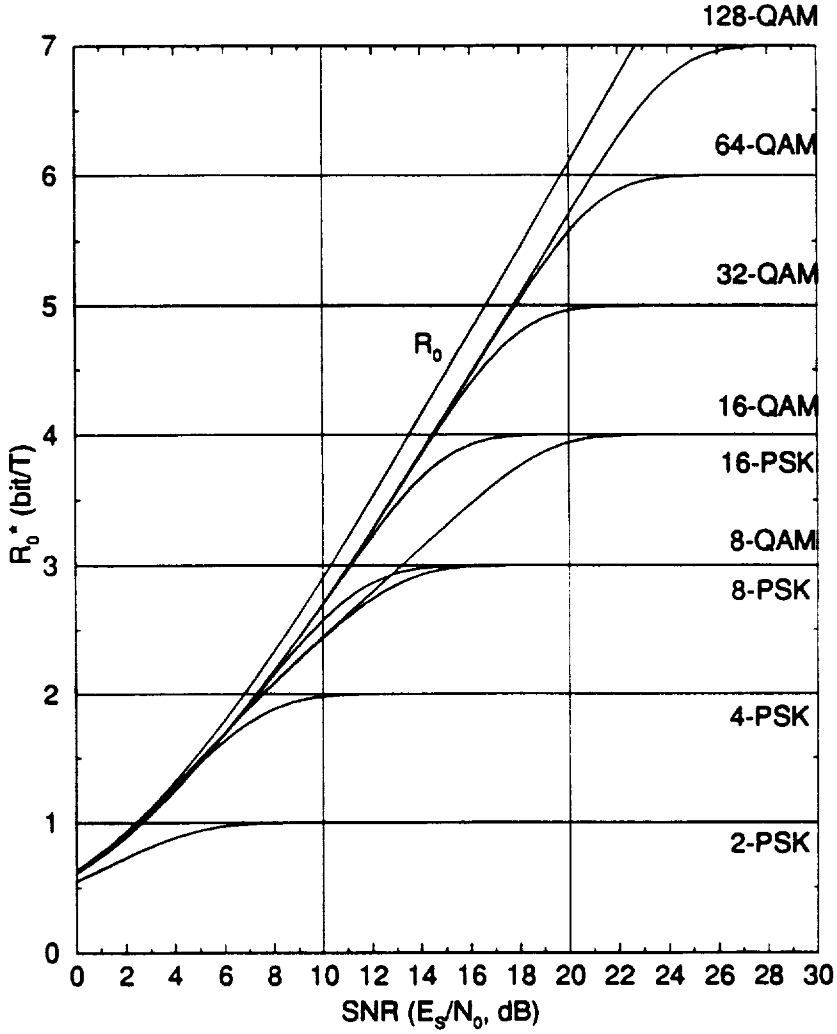


Figure 1.4: Cut-off rate of bandlimited AWGN channels with two dimensional modulation

signaling) and $R_0^* = 2, \cdots, 7$ for QAM r-odulation with equiprobable signaling. The results in Tables 1.1 and 1.2 are very similar.

From Figure 1.4, we also see that doubling the number of channel signals achieves almost all the coding gain (in terms of channel cut-off rate) that can be obtained by signal set expansion. This is analogous to the case with channel capacity[70].

Finally, we list the maximum possible shaping gain with respect to channel capac-

Table 1.2: The required SNR to achieve R_0 and R_0^*

R_0 or R_0^* SNR system	2	3	4	5	6	7
256-QAM	7.3	11.1	14.5	17.8	20.9	24.1
128-QAM	7.3	11.1	14.5	17.8	21.0	–
64-QAM	7.3	11.1	14.5	17.9	–	–
32-QAM	7.3	11.1	14.6	–	–	–
16-QAM	7.3	11.2	–	–	–	–
8-QAM	7.4	–	–	–	–	–
Ideal	6.8	10.3	13.5	16.6	19.7	22.7

ity C and channel cut-off rate R_0 in Table 1.3. Note that no shaping gain is available for PSK modulation since the signals in the constellation all have the same energy and thus the average signal energy cannot be reduced with a nonequiprobable signaling.

1.3 Trellis Coded Modulation

Shannon[68] showed that there exist some coding schemes that can achieve the channel capacity and coding can be used to reduce the required signal energy to transmit certain amount of information. The reduction in the signal energy of a coded system

Table 1.3: Maximum possible shaping gain

rate gains respect to	2	3	4	5	6	7
C	0.8	0.8	1.0	1.3	1.3	1.5
R_0	0.8	0.9	1.1	1.3	1.3	1.4

over an uncoded system is called the coding gain. Two fundamental problems in coding theory are then the construction of good codes and the design of efficient decoding algorithms for the known codes to achieve the coding gains promised by the channel capacity. Traditionally, block codes and convolutional codes are used to achieve the coding gains. However, as we explained in Section 1.1, it is impossible for this kind of coding schemes to achieve a spectral efficiency better than 2 bits/T. To achieve better bandwidth efficiency, joint design of coding and modulation is necessary. The Trellis Coded Modulation (TCM) scheme developed by Ungerboeck[70] is such a technique that combines coding and modulation into one scheme to achieve larger bandwidth efficiency. Another technique, which is called the multilevel coding scheme, was originally introduced by Imai and Hirakawa[36] and have been investigated intensively by many others[41, 92]. TCM, which will be referred as trellis codes thereafter, and its decoding will be studied in this dissertation.

The structure of the trellis codes considered in this dissertation is shown in Figure 1.5. To send k information bits/T, a 2^{k+1} point two-dimensional signal constellation is used. The information sequence is divided into k subsequences $\mathbf{x}_j^i$ $(i = 1, 2, \cdots, k)$

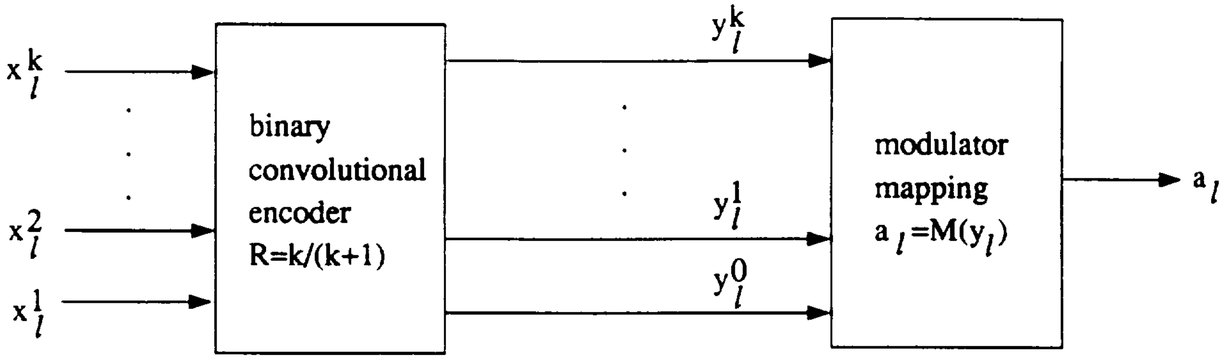

x_l^k
x_l^2
x_l^1
binary convolutional encoder R=k/(k+1)
y_l^k
y_l^1
y_l^0
modulator mapping a_l=M(y_l)
a_l

Figure 1.5: Code structure of trellis codes

and fed into a rate $R = k/(k+1)$ convolutional encoder in systematic feedback form. The encoded (k+1) bits $\mathbf{y}_l = (y_l^k, \cdots, y_l^1, y_l^0)$ are then mapped to a signal point in the 2^{k+1} point signal constellation. Once the mapping is chosen, the performance of trellis codes is determined by the selection of the systematic feedback convolutional code.

Using polynomial notation, code sequences $y(D)$ of a systematic feedback code can be generated by

$$\mathbf{y}(D) = \mathbf{x}(D)\mathbf{G}(D), \tag{1.13}$$

where

$$\mathbf{y}(D) = (y^k(D), \cdots, y^1(D), y^0(D)), \tag{1.14}$$

$$\mathbf{x}(D) = (x^k(D), \cdots, x^2(D), x^1(D)), \tag{1.15}$$

$$\mathbf{G}(D) = \begin{bmatrix} 1 & 0 & \cdots & 0 & H^k(D)/H^0(D) \\ 0 & 1 & \cdots & 0 & H^{k-1}(D)/H^0(D) \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & 1 & H^1(D)/H^0(D) \end{bmatrix}, \tag{1.16}$$

and

$$y^i(D) = y_0^i + y_1^i D + y_2^i D^2 + \cdots\cdots, i = 0, 1, \cdots, k \tag{1.17}$$

$$x^i(D) = x_0^i + x_1^i D + x_2^i D^2 + \cdots\cdots, i = 1, 2, \cdots, k \tag{1.18}$$

$$H^j(D) = h_0^j + h_1^j D + \cdots + h_\nu^j D^\nu, j = 0, 1, \cdots, k. \tag{1.19}$$

$G(D)$ is called the code generator matrix. $H^j = (h_0^j, h_1^j, \cdots, h_\nu^j)$ are the parity-check coefficients associated with the encoder output y^j.

A general implementation of the systematic feedback codes described above is shown in Figure 1.6. If $H^0(D) = 1$ or 0, it results in a class of codes called systematic feedforward codes[49]. All the codes constructed in this paper with a $H^0(D) \neq 1$ or $\neq 0$, i.e., we are only interested in systematic feedback codes since they achieve larger free distances than systematic feedforward trellis codes. The number of memory elements ν in the encoder is called the code constraint length.

Note that some input information bits ($\tilde{k}$ + 1 to k) may be uncoded. In this case, the corresponding $H^j(D)$ ($j = \tilde{k} + 1, \cdots, k$) are equal to zero. Encoders with some uncoded bits simplify code construction and decoding complexity, but limit the achievable free distance for larger constraint lengths. For some short constraint lengths, however, encoders with uncoded bits give optimum free distance codes[70]. The uncoded bits introduce parallel transitions in the code trellis. For $\tilde{k} = 1$, parallel transitions limit the potential asymptotic coding gain to 3.0 dB, while for $\tilde{k} = 2$ and $\tilde{k} = 3$ the potential coding gains are limited to 6.0 dB and 9.0 dB, respectively.

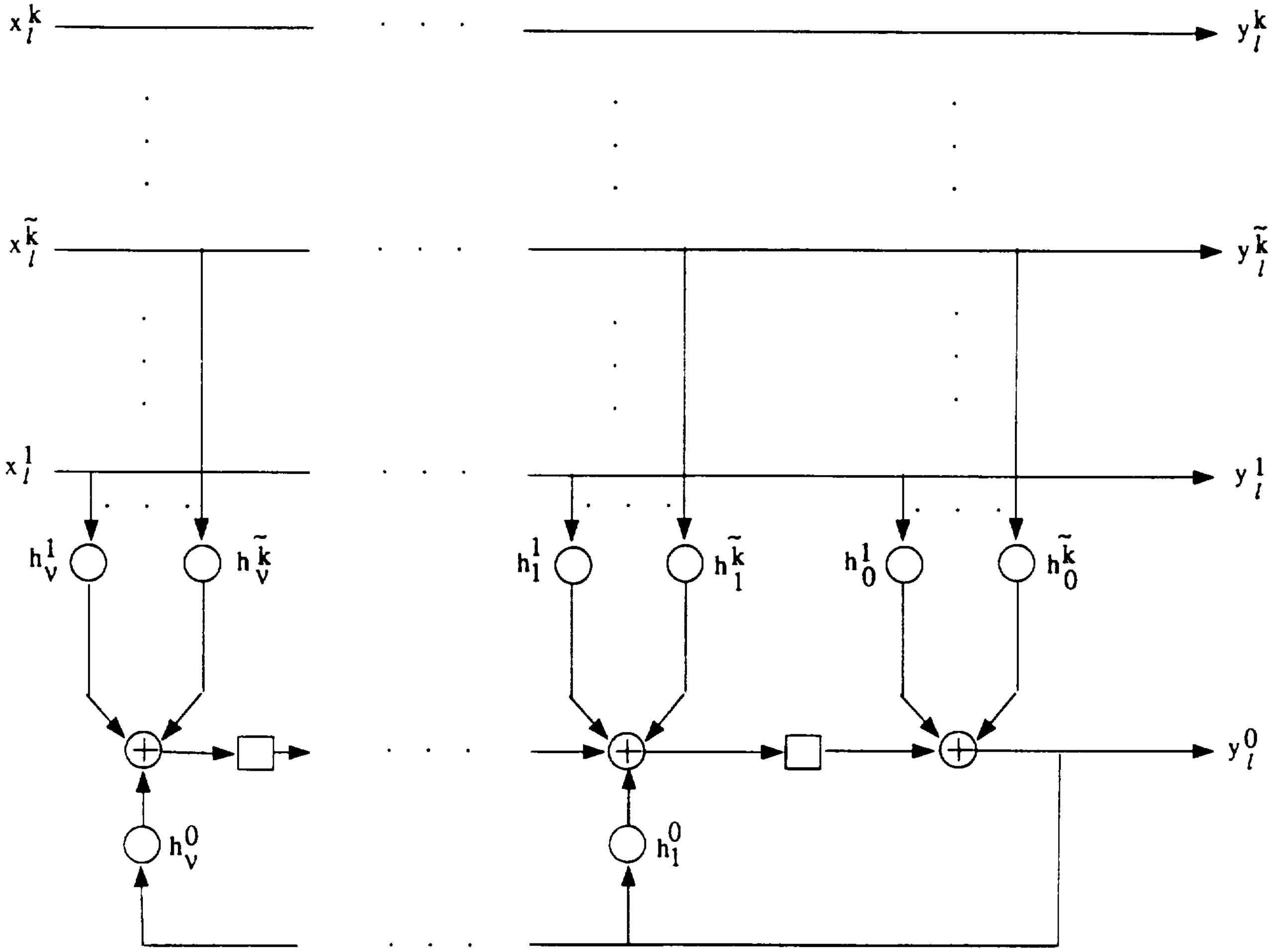


Figure 1.6: General implementation of systematic feedback encoders

The output $\mathbf{y}_l = \{y_l^k, \cdots, y_l^1, y_l^0\}$ is fed into a modulator which transforms the (k+1)-tuple into a two dimensional real number a_l. This one-to-one map is written as

$$a_l = M(\mathbf{y}_l). \tag{1.20}$$

The mapping rule called "mapping by set partitioning" by Ungerboeck[70] is used. This mapping follows from successive partitioning of a constellation into subsets with increasing minimum intra-distances between signals of these subsets.

Now, let us define some of the distance parameters which will be useful in the discussion that follows.

Definition 1.1: The *column distance function (CDF)* of order i for a trellis code, d_i^2, is defined as

$$d_i^2 = \min_{\mathbf{x} \neq \mathbf{x}'} \left[\sum_{l=0}^{i} d^2 \left[M(\mathbf{y}_l), M(\mathbf{y}_l') \right] \right] \tag{1.21}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are two distinct information sequences and $d^2\left[M(\mathbf{y}_l), M(\mathbf{y}_l')\right]$ is the squared Euclidean distance between $M(\mathbf{y}_l)$ and $M(\mathbf{y}_l')$.

Ungerboeck [70] defined the Euclidean weights

$$w^2(\mathbf{e}_l) \triangleq \min d^2[M(\mathbf{y}_l), M(\mathbf{y}_l \oplus \mathbf{e}_l],$$

where $\mathbf{e}_l = [e_l^k, \cdots, e_l^1, e_l^0]$ is an (k+1)-bit error vector and the minimization is over all $\mathbf{y}_l = [y_l^k, \cdots, y_l^1, y_l^0]$, and showed that there always exist a code sequence $(\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_i)$ such that

$$\sum_{l=0}^{i} d^2 \left[M(\mathbf{y}_l), M(\mathbf{y}_l \oplus \mathbf{e}_l \right] = \sum_{l=0}^{i} w^2(\mathbf{e}_l) \tag{1.22}$$

The codes considered in this dissertation are Ungerboeck codes and thus satisfy (1.22), which implies the so-called uniform distance property defined in [2]. The significance of (1.22) is that the column distance function d_i^2 can be calculated by assuming that the all zero sequence is sent. In light of (1.22), the column distance function can also be expressed as

$$d_i^2 = \min_{\mathbf{e}_l \neq \mathbf{0}} \left[\sum_{l=0}^{i} w^2(\mathbf{e}_l) \right] \tag{1.23}$$

i.e., the minimum weight among all the non-zero sequence.

Definition 1.2: $\mathbf{d}^2 = (d_0^2, d_1^2, \cdots, d_\nu^2)$ is called the *distance profile* of a trellis code.

Definition 1.3: $d_{min}^2 = d_\nu^2$ is called the *minimum distance* of a trellis code.

Definition 1.3 follows the convention for convolutional codes[39]. The minimum distance of a convolutional code determines the guaranteed error-correcting capability when the code is decoded by a feedback decoder (threshold decoding) [50]. We notice that d_ν^2 is rather special in its own right even in the case of trellis codes since it can and only can be determined by all the generator (parity-check) coefficients.

Definition 1.4: d_∞^2 is called the *free distance* of the code.

It is seen that a trellis code is determined by its parity-check coefficients $H^j = (h_0^j, h_1^j, \cdots, h_\nu^j)$ $(j = 0, 1, \cdots, \nu)$. Code construction is to select H^j $(j = 0, 1, \cdots, \nu)$ such that some of the distance parameters defined above be optimized.

1.4 Viterbi Decoding and Sequential Decoding

For a given code, we want to have a decoding algorithm that can make as few errors as possible. The Viterbi algorithm[73] is optimum in this sense when equiprobable signaling is used. The Viterbi algorithm was originally introduced by Viterbi[73] to decode convolutional codes. It was recognized to be a maximum likelihood decoding algorithm by Omura[56] and Forney[18, 19]. Forney[17] also pointed out that the Viterbi algorithm could be used to produce the maximum likelihood estimate of the transmitted sequences over a channel with intersymbol interference. Recently, the

Viterbi algorithm has been generalized to provide some kind of reliability information about the decoded sequence by Hashimoto and many others [34, 35, 93].

Before discussing the application of Viterbi decoding and sequential decoding to trellis codes, we describe various methods to represent trellis codes. We note that there are certain number of memory elements in a trellis encoder and thus the encoder can be regarded as a finite state machine. The code can then be described by a finite state diagram with a total number of states 2^{ν} where ν is the code constraint length. For example, the state transition diagram of a constraint length $\nu = 2$ trellis code for 8-PSK modulation taken from [70] can be shown as in Figure 1.7. The state diagram

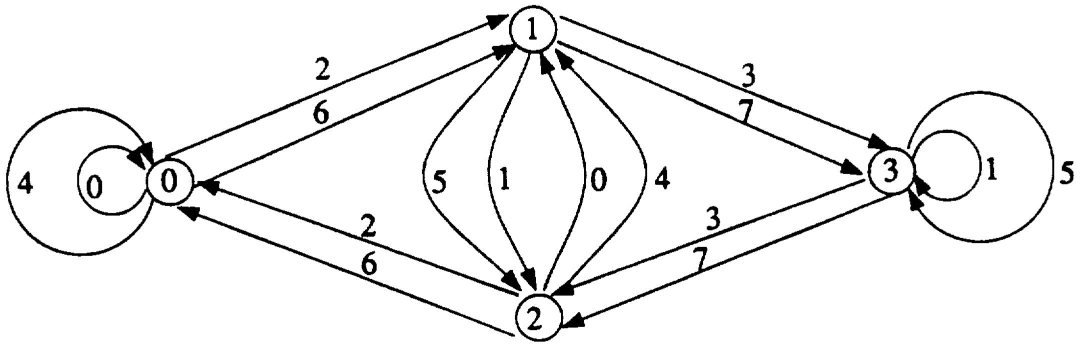


Figure 1.7: State transition diagram for a rate 2/3, 4-state, trellis coded 8-PSK modulation system

is a kind of graph. Omura[56] showed that the Viterbi algorithm was equivalent to a dynamic programming solution to the problem of finding the shortest path through a weighted graph. The distance properties of trellis codes can be analyzed using the state transition diagram[95].

To explain the operations of the Viterbi algorithm, we can expand the 2^{ν} state diagram in the time unit and get a 2^{ν} state trellis diagram. For example, the state diagram of Figure 1.7 can be expanded into a trellis diagram as shown in Figure 1.8. The trellis corresponds to a information sequence of $L = 5$ branches and a one

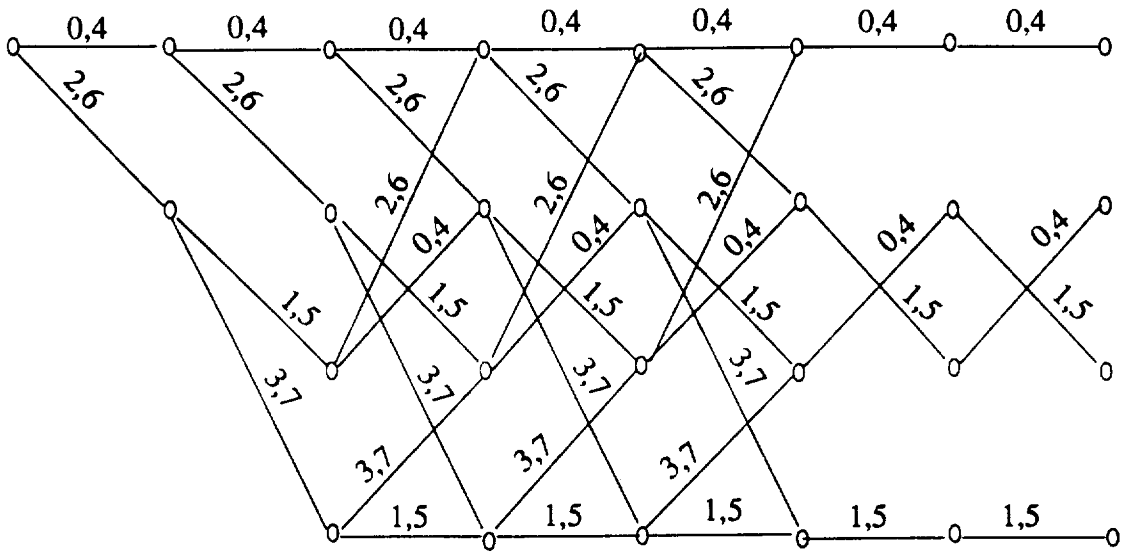


Figure 1.8: Trellis diagram for a rate 2/3, 4-state, trellis coded 8-PSK modulation system

constraint length all-zero tail. It is noted that the trellis cannot return to all-zero state. This is a common feature of convolutional codes in feedback form.

The Viterbi algorithm searches through the trellis and select one survivor path associated with each state. Thus, the computational complexity of the Viterbi algorithm is primarily determined by the number of states, which is 2^ν for a constraint length ν code. We define the operations that are required to select one survivor as one computation. Thus, it requires about 2^ν computations to decode one branch. We note that the number of computations required to decode one branch for a Viterbi decoder grows exponentially with ν.

For codes in feedforward form, the trellis will merge into the all-zero state if a one constraint length all-zero tail is added at the end of information sequence. In this case, only one path survives at the end of decoding. However, for trellis codes that are constructed in systematic feedback form, the trellis will not merge into the all-zero state as shown in Figure 1.8 and in fact there still are 2^{ν} survivor paths when the $L+\nu$ branches are decoded. In this case, we may select the path that has the best metric as the decoded path. Normally, this provides less error protection for the information bits at the end of the sequence and thus more errors will be caused. However, if L is very large, this effect is negligible.

Sequential decoding was shown to be a good alternative to the Viterbi algorithm for convolutional codes. It was first suggested by Wozencraft [91] to decode convolutional code. In 1963, Fano [14] introduced a new version of sequential decoding, which is now called the Fano algorithm. Several years later, Zigangirov [96] and Jelinek[38] independently discovered another version of sequential decoding, which is called the stack algorithm. A variety of variations have also been suggested [7, 32]

A sequential decoder can operate in two modes: block decoding and continuous decoding. In the block decoding mode, each of the k information subsequences is divided into blocks of finite length L. The encoding of each block starts from the all zero (or some other known) state. Usually, a one constraint length tail of ν all zero (or some other predetermined) bits follows each information block to guarantees good performance. Otherwise ($L \to \infty$), the decoder is operated in a continuous decoding mode.

There are 2^{kL} codewords of length $n(L+\nu)$ for a rate k/n trellis code operating in the block decoding mode with constraint length ν and information sequence length kL. In discussing sequential decoding, it is convenient to represent these codewords as paths through a code tree containing $L+\nu+1$ time units or levels. Every path in the tree is distinct from every other path. An example of the code tree is shown in Figure 1.9 for a constraint length $\nu = 3$ (8-state) trellis code with 8-PSK modulation taken from [70], where $L = 2$.

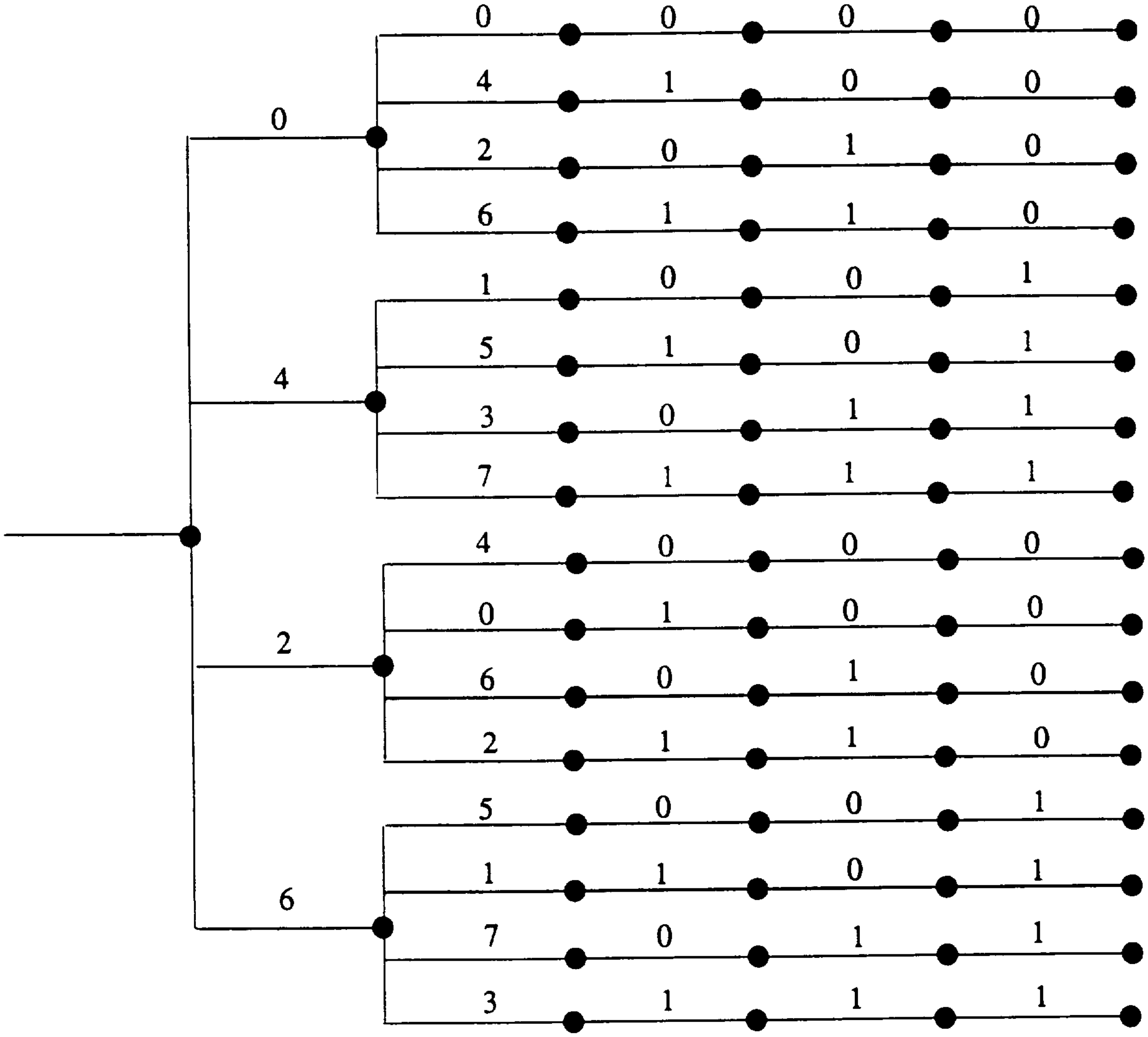


Figure 1.9: Code tree for a rate 2/3, 8-state, trellis coded 8-PSK modulation system

A sequential decoder moves through a code tree. It moves along the correct path in the tree as long as the metric of that path keeps increasing. This feature makes it faster than a Viterbi decoder, which is a kind of exhaustive search through all the paths regardless of their metrics. Actually, it has been shown that the computational effort of sequential decoding is independent of the code constraint length ν[37, 67]. Thus, large constraint length codes can be used to achieve large coding gains using sequential decoding. However, it is quite obvious that the computational effort of sequential decoding depends on the received sequence. If the received sequence is very close to the correct sequence, the decoder will easily follow through the correct path in the code tree. Otherwise, it will encounter some difficulty. This makes the computational effort of sequential decoding a random variable.

The error performance of sequential decoding, on the other hand, is not optimum, but they are very close to the Viterbi algorithm. It has been shown by a random coding approach that sequential decoding can perform almost as well as the Viterbi algorithm[76, 94]. Thus, a slightly larger constraint length code using sequential decoding will outperform a smaller constraint length code using Viterbi decoding with no or little computational penalty.

Finally, we compare the performance of trellis codes using Viterbi decoding with the channel capacity and cut-off rate bounds via simulation. Constraint length of $\nu = 6$ trellis codes are used since we believe that a trellis code with $\nu = 6$ and Viterbi decoding is commercially practical. First, a rate 2/3 Ungerboeck code of constraint length $\nu = 6$ with 8-PSK modulation with Viterbi decoding is simulated. The results along with the channel capacity and cut-off rate bounds are shown in Figure 1.10. It is observed that the code is about 1.4 dB away from the R_0^* bound and about 3.1 dB away from the C^* bound for 8-PSK modulation at a BER of 10^{-5}. Note that the number of computations required to decode one branch is 64. In the rest of the dissertation, we will show that more than one dB coding gain can be achieved at a BER of 10^{-5} with much less computational effort when sequential decoding is used. We will also construct large constraint length trellis codes that can achieve the channel cut-off rate bound for use with sequential decoding.

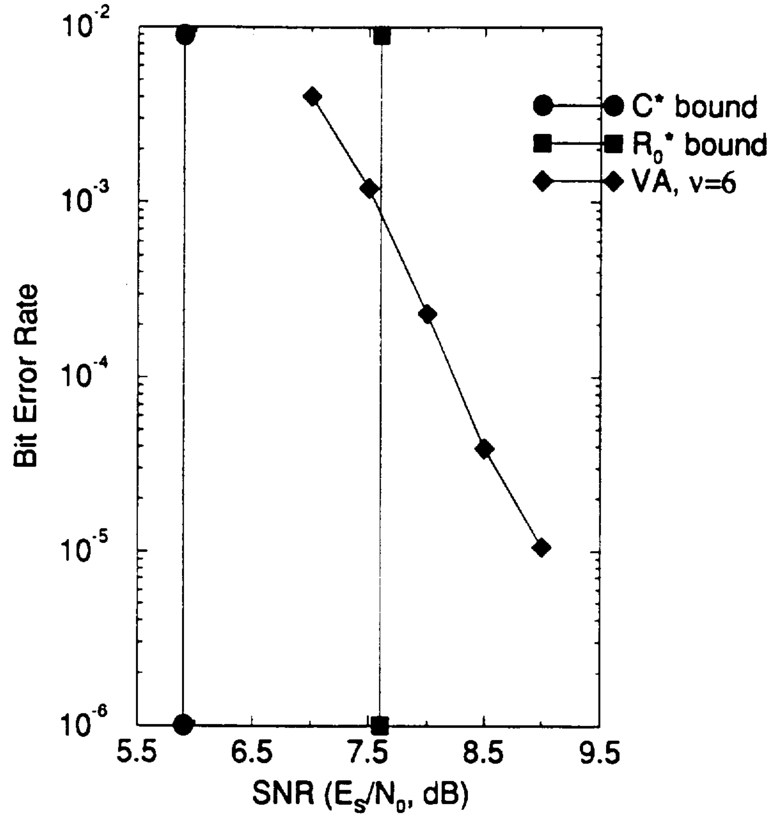


Figure 1.10: Performance of a rate 2/3 trellis coded 8-PSK with $\nu = 6$

Similarly, in Figure 1.11, we compare the performance of a rate 3/4 Ungerboeck code of constraint length $\nu = 6$ with 16-QAM modulation using Viterbi decoding with the channel capacity and channel cut-off rate bounds. In Figure 1.11, the capacity C and the cut-off rate R_0 bounds at a spectral efficiency of 3 bits/T are also shown since they are the bounds achievable when shaping coding is used for QAM modulations. It is also seen that the code is about 1.5 dB away from the R_0^* and about 3.4 dB away from the C^* bound. The application of sequential decoding to trellis coded 16-QAM and the construction of trellis codes with 16-QAM modulation for use with sequential decoding will also be studied in this dissertation.

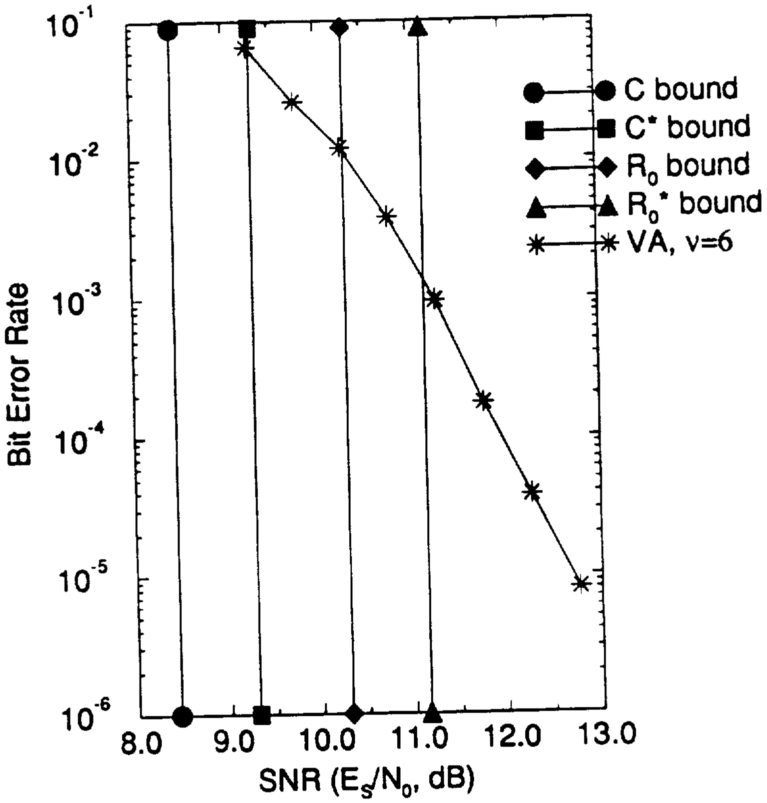


Figure 1.11: Performance of a rate 2/3 trellis coded 16-QAM with $\nu = 6$

1.5 Overview of the Dissertation

This dissertation treats three aspects of trellis coding. Chapters 2 and 3 investigate the application of sequential decoding and its modifications to trellis codes. In Chapters 4 and 5, code construction algorithms are proposed and trellis codes for use with sequential decoding are constructed. In Chapter 6, the relationship between shaping and coding is explored.

Trellis codes can be implemented in systematic feedforward, systematic feedback, and non-systematic feedforward forms. Only systematic feedback and non-systematic feedforward encoders are capable of generating optimum free distance codes. In general, there are many non-systematic feedforward encoders which can generate a given convolutional code. An encoder is minimal if it requires the fewest number of memory elements needed to generate a code[16]. In order to find a minimal encoder, it is always possible to convert a non-systematic feedforward encoder to an equivalent systematic feedback encoder[16, 60]. The systematic feedback encoder is unique, minimal [16], and can never be catastrophic [54]. Also, rate k/(k+1) encoders in systematic feedback form reduce the computer search complexity in constructing trellis codes since a single parity check polynomial determines a code in this form. Thus, most trellis codes are constructed in systematic feedback form. In Chapter 2, the application of sequential decoding to rate k/(k+1) systematic feedback trellis codes is investigated. The relationship between the Fano metric and maximum likelihood decoding is discussed and the Fano metric is derived. The Fano algorithm is briefly reviewed. After the discussions of the demodulator quantization, signal mapping in the tail of a block, and the influence of tail length on performance, simulation results on the error performance and computational effort for trellis codes using sequential decoding are presented.

Computational effort of sequential decoding is a random variable with a Pareto distribution. Although the undetected error probability can be made arbitrarily small, some data cannot be completely decoded and the probability of incomplete decoding (erasure) is usually on the order of 10^{-2} to 10^{-3} [43]. Thus, the performance of sequential decoding is limited in the case where a feedback channel is not available. In Chapter 3, an erasurefree sequential decoding algorithm is introduced. Several versions of the algorithm can be obtained by choosing certain parameters and selecting a resynchronization scheme. These can be categorized as block decoding or continuous decoding, depending on the resynchronization scheme. The performance of a typical block decoding scheme is analyzed. A general resynchronization scheme for continuous sequential decoding is presented. The performance of continuous sequential decoding using this scheme is studied via simulation and the performance of both block decoding and continuous decoding is compared with the VA.

Most of the trellis codes constructed thus far have been for use with the Viterbi algorithm[70, 72]. The asymptotic error performance of the Viterbi algorithm[73] is determined by the minimum free Euclidean distance of the code. Thus, the free distance has been used as the main criterion in code construction for use with the Viterbi algorithm[70, 72]. In Chapter 4, the influence of distance parameters on computational effort of sequential decoding of trellis codes is studied and trellis codes for use with sequential decoding are constructed. First, the influence of the column distance function and distance profile of trellis codes on the computational effort of sequential decoding is studied by analysis and simulation. Trellis codes with Optimum Distance Profiles (ODP) and Optimum Free Distances (OFD) are then constructed and the design criterion for trellis codes with sequential decoding is examined. A new algorithm to construct robustly good trellis codes is presented. The new codes obtained by this approach are compared with the ODP and the OFD codes as well as the best known trellis codes in terms of free distance and distance profile. Simulation results using sequential decoding are also presented to compare the performance of the new codes with the best known codes.

Trellis codes constructed in Chapter 4 have optimum or nearly optimum distance parameters for use with sequential decoding. They are obtained in an exhaustive search with some rejection rules. Thus, it is impossible to construct trellis codes long enough to exploit the computational advantage of sequential decoding. In Chapter 5, probabilistic construction algorithms are investigated for constructing good long trellis codes that can achieve the channel cut-off rate at a Bit Error Rate (BER) of $10^{-5} - 10^{-6}$. The algorithms are motivated by the random coding bound for trellis-type codes. First, results from random coding are reviewed and simulation results for trellis codes are presented to illustrate how randomly chosen codes perform. Two code construction algorithms are then proposed. The two construction algorithms are compared and trellis codes with 8-PSK and 16-QAM modulation are constructed by one of the proposed algorithms. These codes are compared with the best known codes for short constraint lengths. New codes are decoded by the conventional Fano algorithm and the erasurefree sequential decoding scheme proposed in Chapter 3. Performance is compared with uncoded systems and the cut-off rate bound.

A shape gain can be achieved using either higher dimensional spherical constellations[5, 22, 27] or appropriately designed shaping codes [3, 4, 24, 48]. The ultimate (potential) shaping gain is 1.53 dB. It has been shown that a large portion of this gain can be achieved using simple shaping code[4, 24]. However, shaping gain is usually measured in a shaped only system by assuming that shaping and coding are separable. In Chapter 6, the separability of shaping and coding in a coded/shaped modulation system is examined in the context of the achievability of Shannon's bound and additivity of shaping gain and coding gain.

2

SEQUENTIAL ᑐECODING OF TRELLIS CODES

The publication of Ungerboeck's seminal paper[70] on trellis coded modulation stimulated wide interest in the construction of good trellis codes[5, 20, 21, 23, 26, 41, 49, 58, 59, 61, 72, 77, 80, 82, 86, 87, 88]. However, very few papers have addressed the decoding problem [63, 78, 79, 82]. Most researchers assume that the Viterbi Algorithm (VA)[73] is used for decoding and trellis codes are then constructed by hand or by computer search to maximize the minimum free Euclidean distance and minimize the number of nearest neighbors. However, since both the hardware complexity and the computational effort of the VA increase exponentially with the constraint length ν, it is not practical to implement for large ν and its performance is limited to moderate bit error rates (BER's). To achieve better performance requires the use of larger constraint lengths and suboptimum decoding.

It is well known that the computational effort and the hardware complexity of Sequential Decoding (SD) algorithms [7, 14, 32, 38, 45, 78, 79, 82, 91, 96] are essentially independent of the constraint length ν, so large ν can be used and arbitrarily small error probability can be obtained with tolerable complexity and decoding speed. In this chapter, the application of sequential decoding to trellis codes is investigated.

An input buffer is needed in a sequential decoder since its computational effort is a random variable. An infinite buffer is assumed throughout this chapter, i.e., complete decoding is allowed. Sequential decoding with a finite buffer will be discussed in Chapter 3. In Section 2.1, the relationship between the Fano metric and maximum likelihood decoding is explored. The Fano metric for discrete input and continuous output AWGN channel is derived. In Section 2.2, the Fano algorithm is briefly reviewed. In Section 2.3, several quantization schemes are studied via simulation for PSK constellations. In Section 2.4, a simple method to increase the distance of trellis codes at the tail is presented. In Section 2.5, the influence of the tail on performance is studied. In Section 2.6, error performance of trellis codes using sequential decoding is studied via simulation. In Section 2.7, simulation results for computational distribution of sequential decoding of trellis codes are presented.

2.1 Maximum Likelihood Decoding and the Fano Metric

Referring to Figure 1.1, we see that in a trellis coded modulation system with an AWGN channel, the information sequence $\mathbf{x}$ is transformed into a modulated signal sequence $\mathbf{a}$. It is the sequence $\mathbf{a}$ that is transmitted through the channel. The decoder then must produce an estimate $\hat{\mathbf{a}}$ of the modulated sequence $\mathbf{a}$ based on the received sequence $\mathbf{z}$ which is corrupted by an additive white Gaussian noise. From $\hat{\mathbf{a}}$, we can get an estimate $\hat{\mathbf{x}}$ of the information sequence $\mathbf{x}$ since there is a ont-to-one correspondance between the information sequence $\mathbf{x}$ and the modulated signal sequence $\mathbf{a}$. Clearly, $\hat{\mathbf{x}} = \mathbf{x}$ if and only if $\hat{\mathbf{a}} = \mathbf{a}$. Given that $\mathbf{z}$ is received, the conditional error probability of the decoder is defined as

$$P(E|\mathbf{z}) = P(\hat{\mathbf{a}} \neq \mathbf{a}|\mathbf{z}). \tag{2.1}$$

The error probability of the decoder is then given by

$$P(E) = \int P(E|\mathbf{z})P(\mathbf{z})d\mathbf{z}. \tag{2.2}$$

$P(\mathbf{z})$ is independent of the decoder since $\mathbf{z}$ is produced prior to decoding. Hence, an optimum decoder must minimize $P(E|\mathbf{z}) = P(\hat{\mathbf{a}} \neq \mathbf{a}|\mathbf{z})$ to minimize the error probability. Since minimizing $P(\hat{\mathbf{a}} \neq \mathbf{a}|\mathbf{z})$ is equivalent to maximizing $P(\hat{\mathbf{a}} = \mathbf{a}|\mathbf{z})$, $P(E|\mathbf{z})$ is minimized for a given $\mathbf{z}$ by choosing $\hat{\mathbf{a}}$ as the code sequence $\mathbf{a}$ which maximizes

$$P(\mathbf{a}|\mathbf{z}) = \frac{P(\mathbf{z}|\mathbf{a})P(\mathbf{a})}{P(\mathbf{z})}, \tag{2.3}$$

that is, $\hat{\mathbf{a}}$ is chosen as the most likely modulated signal sequence given that $\mathbf{z}$ is received. If all information sequences, and hence all the modulated signal sequences are used with equal probability, maximizing (2.3) is equivalent to maximizing $P(\mathbf{z}|\mathbf{a})$. If the information bits from the source are of equal probability, the resulted information sequences of the same length will have the same probability. Thus, for a conventional coded system (without shaping), the signal sequence will also be of equal probability.

A discrete channel is said to be memoryless if the received signal depends only on the corresponding transmitted signal. For a discrete memoryless channel (DMC), we have

$$P(\mathbf{z}|\mathbf{a}) = \prod_{l} P(z_l|a_l) \tag{2.4}$$

according to the definition. A decoder that chooses an estimate to maximize (2.4) is called a maximum likelihood decoder. The strategy for choosing an estimated code sequence to maximize (2.4) is called the maximum likelihood decoding rule. Since $\log x$ is a monotone increasing function of x, maximizing (2.4) is equivalent to maximizing the log-likelihood function

$$\log P(\mathbf{z}|\mathbf{a}) = \sum_{l} \log P(z_l|a_l). \tag{2.5}$$

We call the log-likelihood function the log-likelihood metric in a decoding algorithm. Obviously, the log-likelihood metric is optimum for the comparison of the sequences with the same length. Using (1.3) and (2.5), we obtain the log-likelihood metric for an AWGN channel

$$\log P(\mathbf{z}|\mathbf{a}) = \sum_{l} -\alpha|z_l - a_l|^2 + \beta, \tag{2.6}$$

where α and β are constants independent of z_l and a_l. Thus, for trellis codes on an AWGN channel, a maximum likelihood decoder chooses an estimate to minimize the Euclidean distance between the received sequence and the estimated sequence. In the Viterbi algorithm, the sequences (partial paths) being compared are always of the same length and thus the log-likelihood metric, or equivalently, the Euclidean distance, can be used as the metric. It has been shown [18, 19] that the Viterbi algorithm can always find an estimated sequence $\mathbf{a}$ that maximizes $\log P(\mathbf{z}|\mathbf{a})$ for a given $\mathbf{z}$ if the log-likelihood metric is used.

We have shown above that the log-likelihood metric can be used to minimize the error probability in a decoder where code sequences of the same length are compared, such as the Viterbi algorithm. However, sequential decoding always involves the comparison of code sequences of different lengths in the decoding process. Since $P(z_l|a_l)$ is always less than one, $\log P(z_l|a_l) = -\alpha|z_l - a_l|^2 + \beta$ is always negative. Thus, if two sequences of different lengths are compared, the shorter one will be favored since it has a larger metric. Similar conclusion can be drawn if the Euclidean distance is used as the metric since the shorter path has a smaller distance. The log-likelihood metric is then no longer optimum in comparison of different length sequences.

In order to derive an optimum metric for sequential decoding, we need to determine $P(\mathbf{a}|\mathbf{z})$ or equivalently $P(\mathbf{a},\mathbf{z})$. Here $\mathbf{a}$ may not have the same length as the received sequence $\mathbf{z}$ since in quential decoding the paths being compared are usually with different lengths provided the same received sequence $\mathbf{z}$ is known. Now let us derive $P(\mathbf{a},\mathbf{z})$ and then the optimum metric for comparison of code sequences of different lengths using a similar approach to Massey's[51].

Consider a variable length code $\{\mathbf{a}_1,\mathbf{a}_2,\cdots,\mathbf{a}_M\}$ whose code lengths are $\{n_1,n_2,\cdots,n_M\}$. The message m $(1 \le m \le M)$, having probability P_m, selects the codeword $\mathbf{a}_m = [a_1^m,a_2^m,\cdots,a_{n_m}^m]$. We add a "random tail" $\mathbf{t}_m = [t_1,t_2,\cdots,t_{N-n_m}]$ to form the input codeword

$$\mathbf{c} = [c_1,c_2,\cdots,c_N] = [\mathbf{a_m},\mathbf{t_m}] \tag{2.7}$$

for transmission over the discrete memoryless AWGN channel. Here $N = \max(n_1,n_2,\cdots,n_M)$ is the maximum codeword length. We assume that $\mathbf{t}_m$ are selected independently of $\mathbf{a}_m$, and that the signals in $\mathbf{t}_m$ is selected independently according to a probability measure $Q()$ over the channel input signal points in the modulation constellation, i.e.,

$$P(\mathbf{t_m}|\mathbf{a_m}) = P(\mathbf{t_m}) = \prod_{k=1}^{N-n_m} Q(t_k). \tag{2.8}$$

Let $\mathbf{z} = [z_1,z_2,\cdots,z_N]$ be the received signal sequence corresponding to $\mathbf{c} = [c_1,c_2,\cdots,c_N]$. We have

$$P(\mathbf{z}|\mathbf{c}) = \prod_{l=1}^{n_m} P(z_l|a_l^m) \prod_{j=1}^{N-n_m} P(z_{n_m+j}|t_j), \tag{2.9}$$

where $P(|)$ denotes the channel transition probability. For a continuous-valued output channel, we may write

$$P(z_l|a_l) = \Delta z_l \times p\{z_l|a_l\}, \tag{2.10}$$

where $p\{z_l|a_l\}$ is the transition probability density function as defined in (1.3). The joint probability of sending message m, adding the tail $\mathbf{t}_m$, and receiving $\mathbf{z}$ can be written as

$$\begin{aligned} P(m, \mathbf{t_m}, \mathbf{z}) &= P_m P(\mathbf{t_m}|\mathbf{a_m}) P(\mathbf{z}|\mathbf{a_m}\mathbf{t_m}) \\ &= P_m \prod_{l=1}^{n_m} P(z_l|a_l^m) \prod_{k=1}^{N-n_m} Q(t_k) \prod_{j=1}^{N-n_m} P(z_{n_m+j}|t_j). \end{aligned} \tag{2.11}$$

Note that there is a one-to-one correspondence between $\mathbf{a}_m$ and m. Summing over all possible random tail, we obtain

$$\begin{aligned} P(\mathbf{a_m}, \mathbf{z}) &= P(m, \mathbf{z}) \\ &= P_m \prod_{l=1}^{n_m} P(z_l|a_l^m) \prod_{j=1}^{N-n_m} P_0(z_{n_m+j}). \end{aligned} \tag{2.12}$$

where

$$P_0(z_l) = \sum_{t_k} P(z_l|t_k) Q(t_k) \tag{2.13}$$

is the probability measure induced on the channel output signals when the channel inputs are used according to $Q()$.

Now according to the maximum likelihood decoding rule, we have to maximize $P(m, \mathbf{z}) / \prod_{l=1}^{N} P_0(z_l)$. Taking logarithm, we obtain

$$M(\mathbf{a}_m, \mathbf{z}) = \sum_{l=1}^{n_m} \left[\log \frac{P(z_l|a_l^m)}{P_0(z_l)} + \frac{1}{n_m} \log P_m \right] \tag{2.14}$$

which is the metric to be maximized by the optimum decoder. By knowing that a code sequence $\mathbf{a}$ corresponds to a message m, we may drop m in the code sequence $\mathbf{a}_m$ and obtain

$$M(\mathbf{a},\mathbf{z}) = \sum_{l=1}^{n_m} \left[\log \frac{P(z_l|a_l)}{P_0(z_l)} + \frac{1}{n_m} \log P_m \right]. \tag{2.15}$$

$M(\mathbf{a},\mathbf{z})$ is called the Fano metric. A variation of (2.15) was first suggested heuristically by Fano[14] for decoding convolutional codes. The above analysis shows that optimum performance in the sense of minimizing error probability can be achieved when the Fano metric is applied in comparison of variable length sequences.

In order to derive a specific expression of the Fano metric for trellis codes, we define the branch Fano metric as

$$M_B(a_l, z_l) = \log \frac{P(z_l|a_l)}{P_0(z_l)} + \frac{1}{n_m} \log P_m. \tag{2.16}$$

Assume that $\{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_M}\}$ represent all the paths in the code tree that has been explored up to the present by a sequential decoder. (An example of a code tree for a trellis code is shown in Figure 1.9.) Su; ,ose that the channel input signals are taken from a collection of signals $\{a^0, a^1, \cdots, a^{K-1}\}$ with probability $p_i = P(a_l = a^i)$ $(i = 0, 1, \cdots, K-1)$. We then have

$$\begin{aligned} P_0(z_l) &= \sum_{i=0}^{K-1} p_i P(z_l|a^i) \\ &= \Delta z_l \times \sum_{i=0}^{K-1} p_i p\{z_l|a^i\} \end{aligned} \tag{2.17}$$

where $p\{z_l|a^i\}$ is the channel transition probability density function. Assume that the signals in the collection are used with equal probability, i.e., $p_i = 1/K$ for $i = 0, 1, \cdots, K-1$. Next assuming that the information bits are independent and equally likely to be 0's or 1's, we then have the a priori probability that the encoder followed path $\mathbf{a}_m$

$$P_m = 2^{-kn_m} \tag{2.18}$$

for a rate k/n code. Substituting (2.10), (2.17), and (2.18) into (2.16) and noting that $K = 2^n$ and $p_i = 1/K$, we obtain

$$\begin{aligned} M_B(a_l, z_l) &= \log_2 \frac{Kp\{z_l|a_l\}}{\sum_{i=0}^{K-1} p\{z_l|a^i\}} - k \\ &= \log_2 \frac{\exp(-|z_l - a_l|^2/2\sigma^2)}{\sum_{i=0}^{K-1} \exp(-|z_l - a^i|^2/2\sigma^2)} + n(1-R) \end{aligned} \tag{2.19}$$

where $R = k/n$ is the code rate. It is seen that the Fano metric is determined by the received signal z_l and the hypothetical signal a_l.

For a continuous output channel, z_l can be any point in two dimensional Euclidean space. In practice, z_l may be quantized into one of a finite number of values $z_l^{(j)}, j = 1, 2, \cdots, J$. We denote this as $z_l \rightarrow z_l^{(j)}$. There are also many other points that may be quantized into $z_l^{(j)}$. We denote the set of these points as

$$S_l^{(j)} = \{z_l | z_l \rightarrow z_l^{(j)}\}, \quad j = 1, 2, \cdots, J. \tag{2.20}$$

$S_l^{(j)}$ is called the j-th decision region at time l. For memoryless channels, the decision regions are independent of time, i.e., $S_l^{(j)} = S^{(j)}$ for all l. Thus, for quantized outputs, $p\{z_l|a_l\}$ and $p\{z_l|a^i\}$ in (2.19) are replaced by

$$P\{S^{(j)}|a_l\} = \int_{S^{(j)}} p\{x|a_l\} dx \tag{2.21}$$

and

$$P\{S^{(j)}|a^i\} = \int_{S^{(j)}} p\{x|a^i\} dx, \tag{2.22}$$

respectively, when $z_l \rightarrow z_l^{(j)}$.

2.2 The Fano Algorithm

A variety of tree searching algorithms fall into the general category of sequential decoding. Among these, the two most popular algorithms are the stack algorithm ($Z - J$ algorithm)[38, 96] and the Fano algorithm [14]. The stack algorithm uses a stack to store the examined paths. The path with the largest metric is placed in the top of the stack and other paths are stored in decreasing order of metrics. The stack algorithm extends the top path in the stack, by adding its 2^k successors and deleting the top path for a rate k/n code. The paths in stack are then rearranged in oder of decreasing metrics. The decoding stops when the top path reaches the end of the code tree. There are 2^{kL} codewords for a rate k/n code and information sequence of kL. Thus, if the stack depth is smaller than 2^{kL}, the stack may overflow. In practice, the stack depth may always be much smaller than 2^{kL} which is very large even for a moderate L. The Multiple Stack Algorithm of Chevillat and Costello[8] attempts to solve the stack overflow problem. On the other hand, because it does not require any storage, the Fano algorithm does not have a stack overflow problem. In order to insure extending the path with the best metric (the top path), the stack algorithm requires a large effort to continually re-order the stack. This problem can be partially solved by using the stack bucket algorithm, but the Fano algorithm still decodes faster than the stack algorithm for moderate rates [29]. Thus the FA is preferred in most practical implementations [25, 43].

Fano algorithm decoder moves forward or backward from node to node in the code tree depending on whether the cumulative Fano metric at the current node M_C is larger or smaller than some threshold T. T is increased or decreased in steps of some appropriate value Δ called the threshold increment. Suppose that the decoder is at some node of level l and the cumulative Fano metric is $M_C(l)$. For a rate $R = k/n$ trellis code, there are 2^k successors to this node. The decoder computes the 2^k branch

Fano metrics corresponding to the 2^k successors $M_B(i(l))(i(l) = 1, 2, \cdots, 2^k)$ in order of decreasing values. Then, it attempts to move forward to level $l + 1$ along an untried branch with the largest metric. The cumulative metric at next node is then $M_C(l+1) = M_C(l) + M_B(i(l))$ where $M_B(i(l))$ is the largest metric among the untried branches. If $M_C(l + 1)$ is larger than or equal to T, the decoder moves to the node of level $l+1$ and T is increased to the largest possible value not exceeding $M_C(l+1)$ in steps of Δ. Then, the decoder proceeds at level $l + 1$ as at level l. If $M_C(l + 1)$ is smaller than T, the decoder moves back to the node of level $l - 1$. If $M_C(l - 1)$ is larger than T, the decoder attempts to move forward along those untried branches again. If all the branches stemming from node of level $l - 1$ were tried, the decoder moves back to the node of level $l - 2$. The decoder will move forward and backward in this manner until it is forced back to a node for which the value of M_C is smaller than the current threshold T.

When the decoder is forced back to a node for which M_C is smaller than the current threshold, all the paths stemming from this node must contain at least one node for which the metric falls below T. The situation may arise because of a mistake at that node or a preceding node. It may also be caused by the severe channel noise. No matter what, the threshold must be lowered by Δ to allow the decoder to proceed.

After the threshold is reduced, the decoder tries again to move forward. This leads to the decoder to retrace all the previously explored nodes. However, the threshold must not be increased during the process until an unexplored node is reached. Otherwise, the decoder would keep retracing the same path over and over again.

A flowchart of the Fano algorithm is shown in Figure 2.1. The binary variable F is used to control a gate that allows or prevents the threshold from increasing. If $F = 0$, the threshold can be increased at appropriate node. Otherwise, it is prohibited from increasing. $M_C(l)$ is the cumulative Fano metric at the node of level l. $i(l)$ is a counter

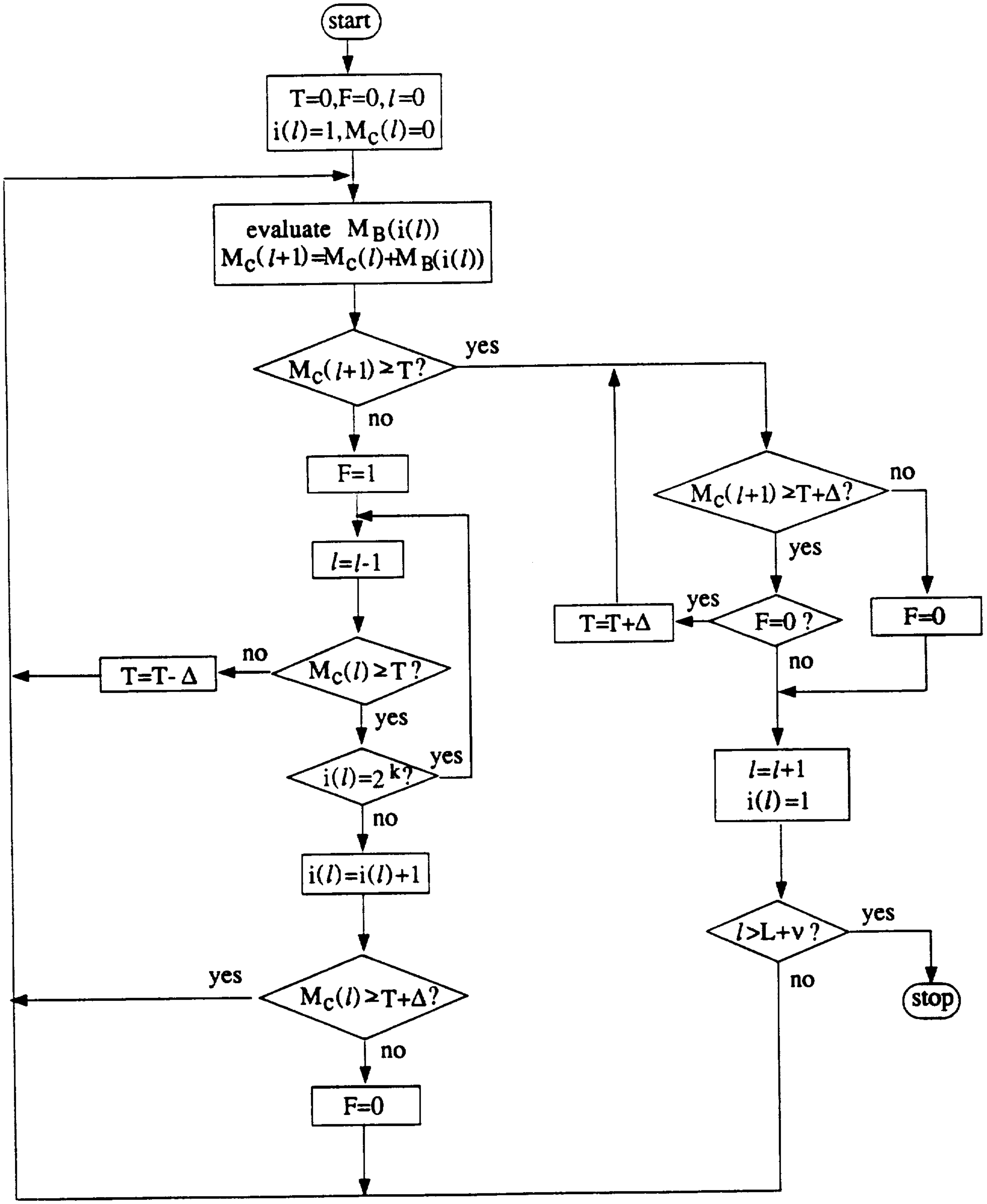


Figure 2.1: A flowchart of the Fano Algorithm

that corresponds to the $i(l)$-th largest Fano metric of the 2^k branches stemming from the node of level l. $M_B(i(l))$ is the $i(l)$-th largest branch Fano metric among all the 2^k branches. ν is the code constraint length and L is the length of information sequence.

The flowchart is self-explanatory. Decoding starts at the original node of level $l = 0$ with the cumulative metric $M_C(l) = 0$, the threshold $T = 0$, $F = 0$, and $i(l) = 1$. Then, the decoder finds out the $i(l)$-th largest branch metric and get the new cumulative metric $M_C(l+1)$ by adding to the metric of the previous level. If the metric $M_C(l+1)$ is less than the threshold T, the decoder moves back to node of level $l-1$. If the metric at this node is smaller than the current threshold, the decoder decreases the threshold by a value of Δ and proceeds from this node. Otherwise, it attempts to move along the branch with the next largest metric. If the metric $M_C(l+1)$ is larger than or equal to the threshold, it moves to the node of level $l+1$ and adjust the threshold if the node has never been visited before. If the new node is the terminal node, decoding stops. Otherwise, decoder proceeds from the new node.

An important feature of the Fano algorithm is that only one path must be stored during the decoding process. This makes the Fano algorithm attractive for practical implementation. Since some nodes may be visited more than once and the number of node visits depends on how severely the signals around this node are corrupted, the number of node visits (computations) required to decode a branch is a random variable. This implies that the computational distribution is an important factor in assessing the performance of a sequential decoder.

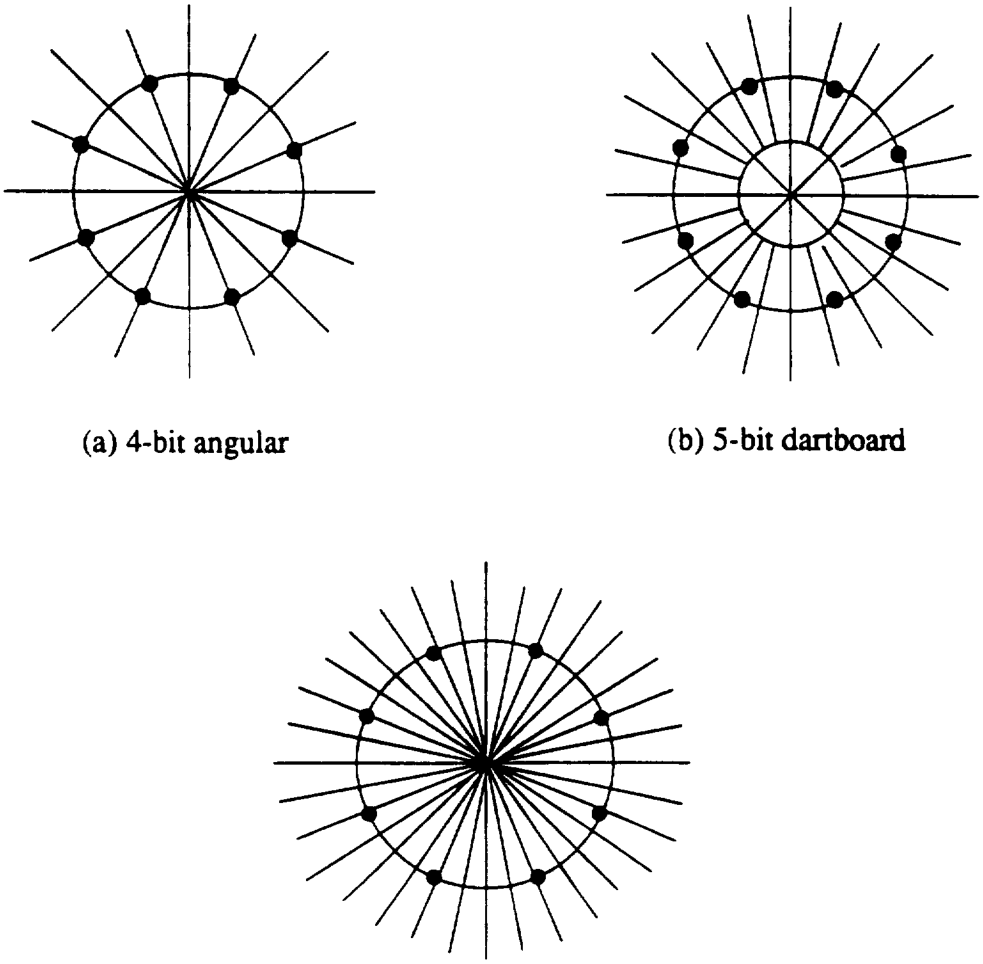
2.3 Demodulator Quantization

The received signal from the channel (demodulator output) z_l can be any two dimensional point in Euclidean space. Since the decoder is digital, z_l must be converted into digital form to be stored and processed. This process is called quantization.

The design of optimal quantizers has been studied in [44, 52, 57, 90]. For M-ary modulation, the demodulator output is quantized into one of $J \geq M$ levels. If $J = M$, the quantizer is said to make hard-decisions. If $J > M$, the quantization is called soft decision. For an equally likely signal set, the nearest neighbor decision rule that assigns the demodulator output to the closest of the M signal points is optimal in the sense of minimizing the error probability for hard decision quantization. In general, it is very difficult to analytically determine the optimum decision regions when the number of quantization levels $J > M$.

Lee [44] established a necessary condition for the boundaries of an optimal J-level quantizer given an arbitrary probability density function relating channel inputs and outputs. Parsons and Wilson[57] showed that the decision regions as shown in Figure 2.2 (a) satisfy Lee's necessary condition for PSK modulations. They further argued that such a quantizer is optimal. However, optimal quantizers for PSK modulations with $J's$ other than M and $2M$ may only be obtained using design algorithms [44, 52, 90]. Figure 2.2 (b) shows another quantizer for 8-PSK modulation proposed in [59] with $J = 32$ levels. This so-called dartboard quantization scheme is obviously not optimum.

Decision regions for soft quantization of PSK modulation may also be rectangular, as shown in Figure 2.3 (a). We have simulated a variety of angular (including the dartboard type of quantization) and rectangular quantization schemes for trellis coded 8-PSK modulation with sequential decoding and found that angular quantization results in the best performance for $J \leq 32$ and that rectangular quantization results in the best performance for $J \geq 64$. The performance of the soft decision decoder using angular quantization improves with increasing J. But very little additional coding gain can be obtained when J exceeds 32. The rectangular quantization performance also improves with J until J reaches 256. In Figure 2.4, we compare the



(a) 4-bit angular

(b) 5-bit dartboard

(c) 5-bit angular

Figure 2.2: Angular quantization schemes for PSK modulation

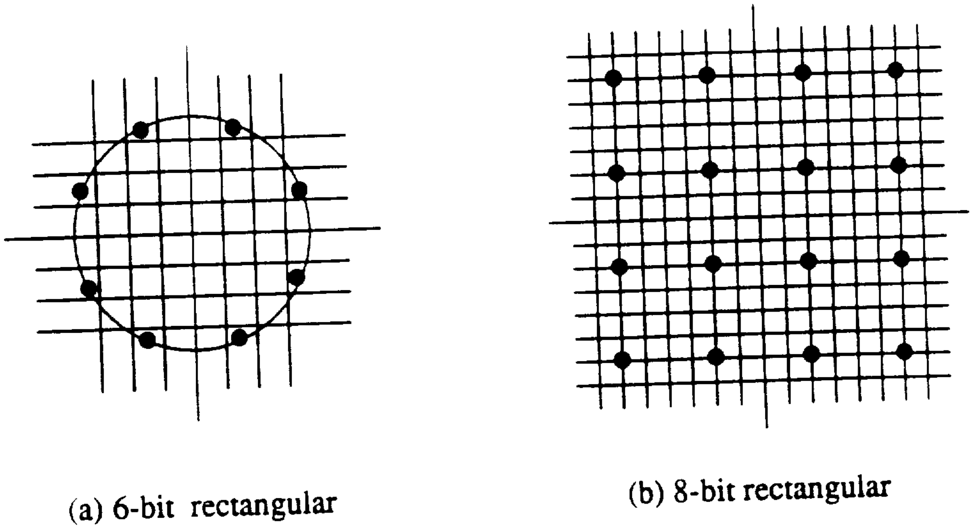


Figure 2.3: Rectangular quantization schemes

BER performance of angular and rectangular quantization for trellis coded 8-PSK modulation using a rate 2/3 Ungerboeck code with constraint length $\nu = 4$[70] and sequential decoding. The 4-bit angular scheme and the 5-bit angular scheme's decision regions are shown in Figure 2.2 (a) and (c). The 6-bit rectangular scheme uses the decision regions shown in Figure 2.3 (a) and similar decision regions are used for the 8-bit rectangular scheme. It is observed that about 0.4 dB more coding gain can be achieved by using the 8-bit rectangular quantization scheme instead of the best angular quantization scheme.

The quantization of QAM modulations is more complicated because of the irregularity of their boundaries. Hueristically, the rectangular quantization schemes shown in Figure 2.3 (b) might be well suited for QAM modulations. It is easy to show that optimal hard decision regions can be obtained using a rectangular quantization scheme. However, for soft QAM modulation quantizers, it is very difficult to deter-

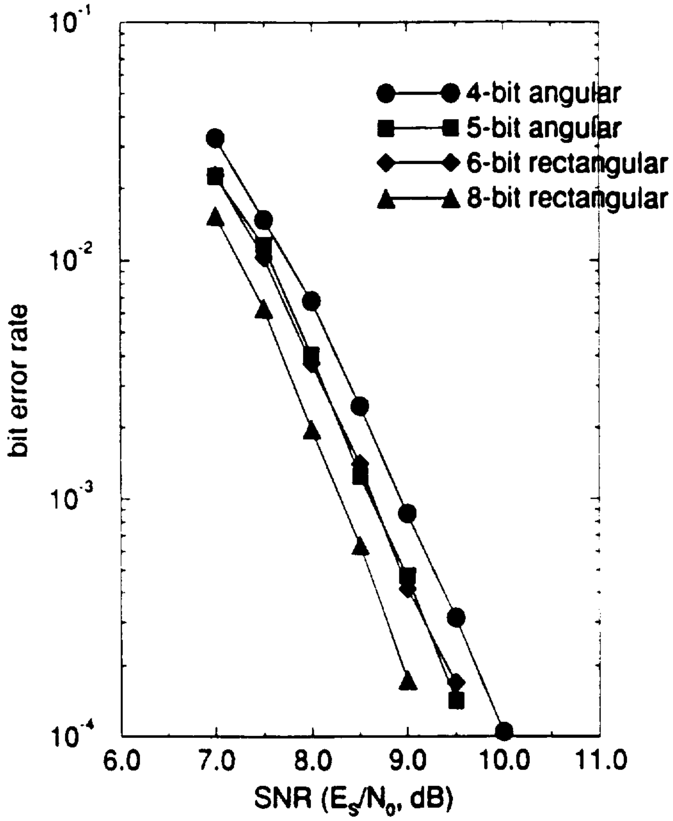


Figure 2.4: Performance of different quantization schemes for trellis coded 8-PSK

mine the decision regions of optimal quantizers. Again, we may optimize the decision regions using design algorithms [44, 52, 90].

2.4 Signal Mapping in the Tail of a Block

We assume that the encoder starts from the all-zero state. In the block decoding mode, the encoder should also terminate in the all-zero state. A one constraint length all-zero tail added to the information sequence can guarantee an all-zero terminal state for feedforward encoders. It has been shown[40] that such a tail is required for sequential decoding of feedforward convolutional codes to maintain good performance.

Since trellis codes are usually constructed in systematic feedback form, the encoder is not returned to the all-zero state by using an all zero tail. This implies that the minimum distance among the encoded signal sequences may be less than the free distance of the code if conventional mapping is used in the tail. We give an example to illustrate this point.

Figure 2.5 shows the natural mapping of 8-PSK channel signals obtained by set partitioning[70], where Δ_0^2, Δ_1^2, and Δ_2^2 are the minimum (squared) subset distances. Note that the distances between 0 (a^0) and 1 (a^1), 0 (a^0) and 2 (a^2), and 0 (a^0) and 4 (a^4) are $\Delta_0^2 = 0.586$, $\Delta_1^2 = 2$, and $\Delta_2^2 = 4$, respectively. In Figure 2.6 (a), three non-zero paths are shown for the Ungerboeck 8-state trellis code with 8-PSK modulation. Paths 1, 2, and 3 represent paths of length $L = 1$ terminated by an all-zero tail. The tail should provide error protection for the information bits at the end of a block. The distance between path 0 and path 2 is $\Delta_1^2 + \Delta_0^2 = 2.586$. By checking the distances of all 6 possible pairs of paths, we find that the distance between path 0 and path 2, i.e., 2.586, is the minimum distance among the four paths. This is much

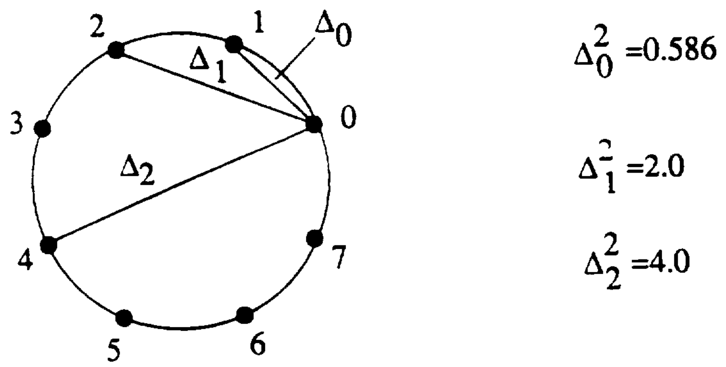
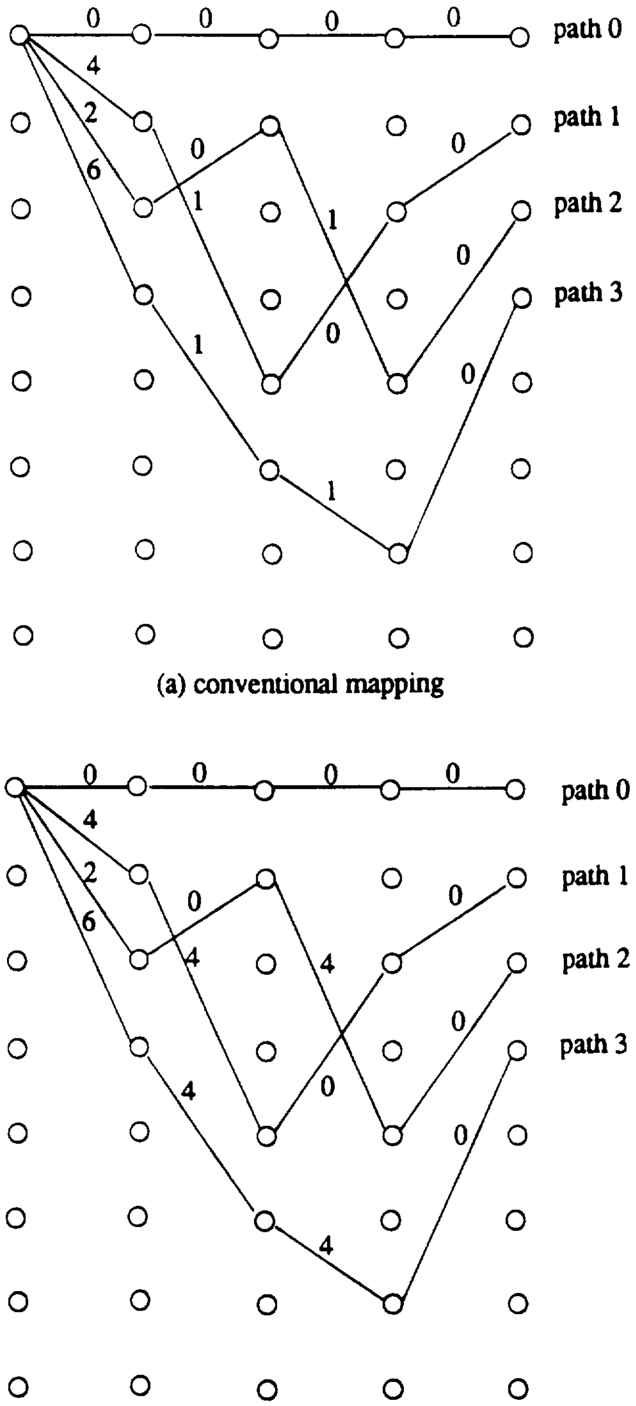


Figure 2.5: Natural mapping for 8-PSK modulation



(a) conventional mapping

(b) maximum distance mapping

Figure 2.6: Alternate tail mappings for trellis coded 8-PSK

smaller than the code free distance of 4.586. This results in less error protection for the information bits at the end of a block. These errors dominate the BER at high SNR, especially for short blocks. However, since there are only two signals in the tail (0 and 1 with conventional mapping), we can change the mapping rule in the tail to maximize the distance between these two signals. For example, we can map the two possible encoder outputs in the tail into 0 and 4, as shown in Figure 2.6 (b). This increases the minimum distance among the four paths to 6.0 and eliminates the reduced error protection for the information bits at the end of a block. Finally, we should point out that the loss of distance in the tail for conventional mapping does not pose a serious problem for continuous decoding algorithms such as Viterbi decoding, since the information sequence is very long and thus the influence of the tail is negligible.

2.5 The Influence of Tail Length on Performance

The influence of the tail length on the error probability of sequential decoding of convolutional codes has been studied in [40]. The undetected block error probability was evaluated as a function of block length and tail length. It was found that the performance improves with increasing tail length until it reaches the constraint length ν. Beyond this, the performance remains nearly the same. We have performed similar simulations for sequential decoding of trellis codes and the BER as a function of the tail length is shown in Figure 2.7, where L is the block length in branches. Optimum Distance Profile (ODP) trellis codes of constraint length $\nu = 10$ and 14 were used [84, 85]. The mapping in the tail followed the approach described in Section 2.4. Our results show that the BER decreases with increasing tail length and that this trend continues until the tail length reaches the constraint length ν. This is consistent with the observation in [40]. For small tail lengths, the information bits at the end of each

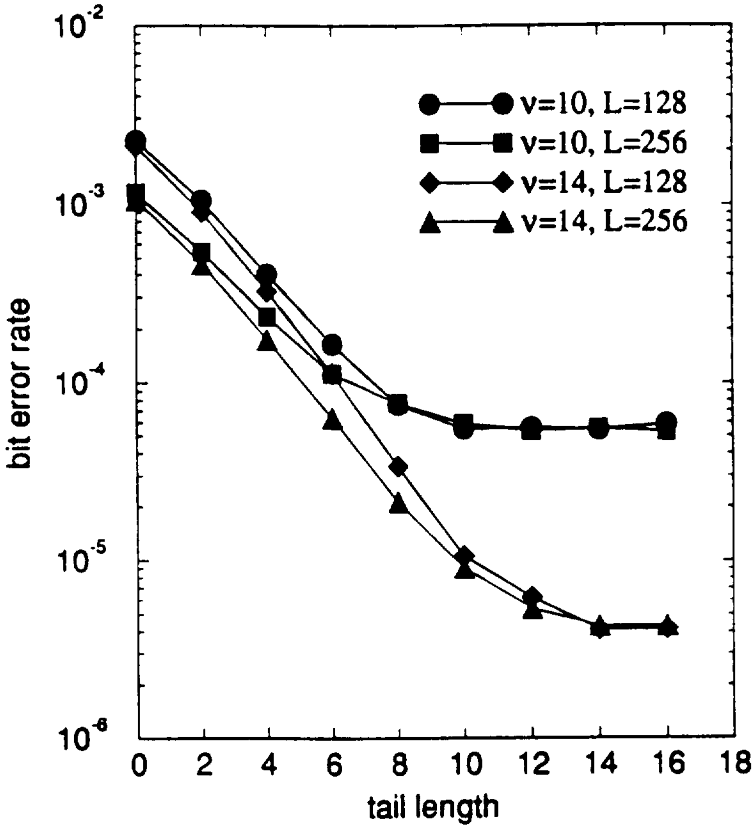


Figure 2.7: Sequential decoding performance vs. the tail length of trellis codes

block get little protection. This effect causes many errors and dominates the BER curves as shown in the figure. We conclude that at least a one constraint length tail is required to minimize the BER.

In [40], it was also observed that the block error probability increases with increasing block length. On the other hand, Figure 2.7 indicates that the BER is independent of the block length as long as the tail length is larger than the constraint length.

2.6 Performance of Sequential Decoding

Sequential decoding has been shown to be a good alternative to the Viterbi algorithm for convolutional codes[46]. It was shown that sequential decoding can perform about as well as the Viterbi algorithm for convolutional codes by analysis and simulations [7, 76, 94]. In [63], Pottie and Taylor compared the block error probability of trellis codes using the Viterbi algorithm, the M-algorithm [45], the Fano algorithm[14], and the generalized stack algorithm[32]. Their results conform closely to previous experience with convolutional codes. In this section, the bit error rate of trellis codes using sequential decoding is compared with the Viterbi algorithm and the channel cut-off rate bound.

Figure 2.8 shows the bit error rate of trellis codes for 8-PSK modulation as a function of the SNR using the Viterbi algorithm and sequential decoding. The Fano Algorithm with a threshold $\Delta = 4.0$ was used in our simulation along with Ungerboeck codes with constraint length $\nu = 6$ and 8[70]. The performance of the Viterbi Algorithm with a constraint length $\nu = 6$ Ungerboeck code is also shown. It shows that the Fano algorithm loses about 0.2 dB coding gain compared with the Viterbi algorithm. However, the Fano algorithm needs a much smaller computational effort. Furthermore, since its computational effort is essentially independent of the code constraint length, sequential decoding can overcome its suboptimum performance compared to Viterbi decoding by using a slightly larger constraint length.

In Figure 2.9, the performance of different constraint length trellis codes with block mode sequential decoding of block length $L = 512$ is shown as a function of SNR. The Fano Algorithm with a threshold $\Delta = 4.0$ was used in the simulations along with ODP trellis codes constructed in [84, 85]. Referring to Figure 1.4, we see that an SNR= 7.6 dB is required to achieve a cut-off rate $R_0^* = 2$ bits/T for an 8-PSK modulation channel. (We refer to the SNR to which the cut-off rate equals a

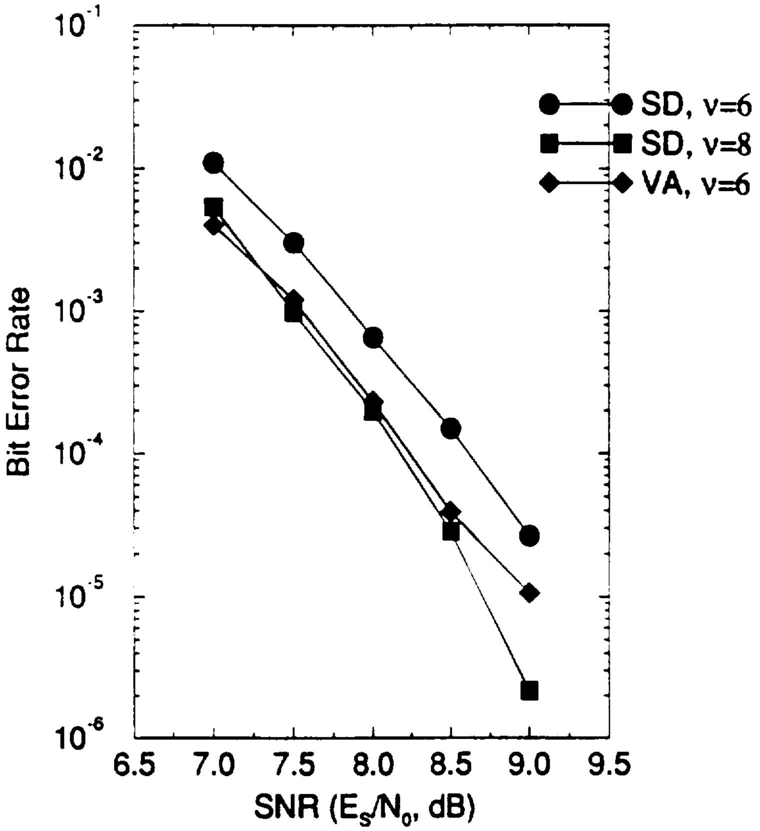


Figure 2.8: Sequential decoding vs. Viterbi decoding

given spectral efficiency as the cut-off rate bound.) The performance of the Viterbi Algorithm with a constraint length $\nu = 6$ code taken from [70] and the cut-off rate bound for 8-PSK modulation at a spectral efficiency of 2 bits/T are also shown in Figure 2.9. We see that sequential decoding provides about a 1.4 dB coding gain over 64 state Viterbi decoding at a BER of 10^{-5} and that the performance gap widens at lower BER's. Since the decoding complexity of sequential decoding is essentially independent of constraint length, this improvement comes without significant increase in decoder complexity. It is also observed that the performance of sequential decoding steadily improves with increasing code constraint length and that the cut-off rate bound can be achieved at a BER of 10^{-5} using a constraint length 15 or longer code.

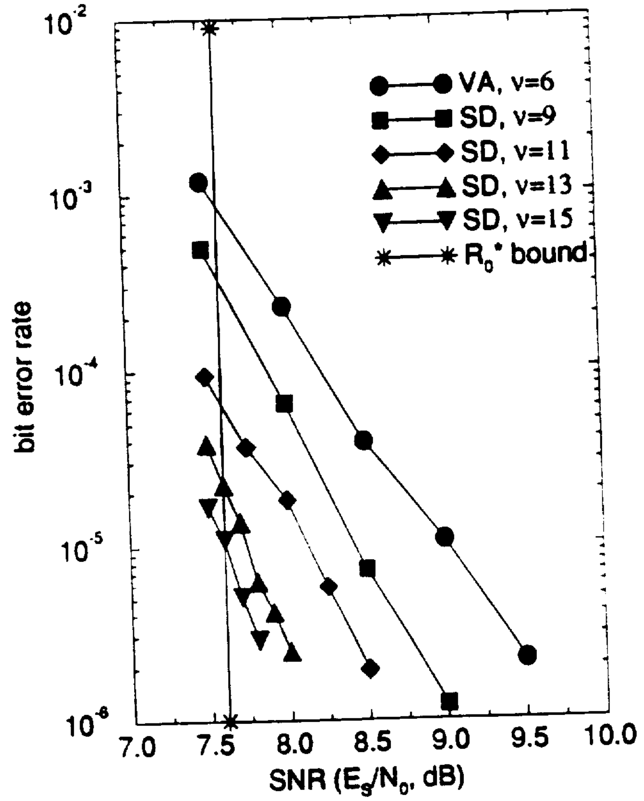


Figure 2.9: Performance of trellis codes using sequential decoding

2.7 Computational Distribution of Sequential Decoding

Define C_b as the number of computations required to decode one branch of the code tree. Suppose that the block length is $L + \nu$. The total number of computations required to decode a block is then $C_B = C_b(L+\nu)$. For Viterbi decoding, the number of computations C_b is fixed and equals to 2^ν for a constraint length ν code. For sequential decoding, C_b is a random variable and so is C_B. It is well known that the computational distribution of convolutional codes with sequential decoding can be approximated by a Pareto distribution [37, 67], i.e.,

$$Pr(C_b > N) = AN^{-\rho}, \tag{2.23}$$

where A and ρ are constants related to the specific code and the specific version of sequential decoding used and the channel characteristics. In Figure 2.10, we plot

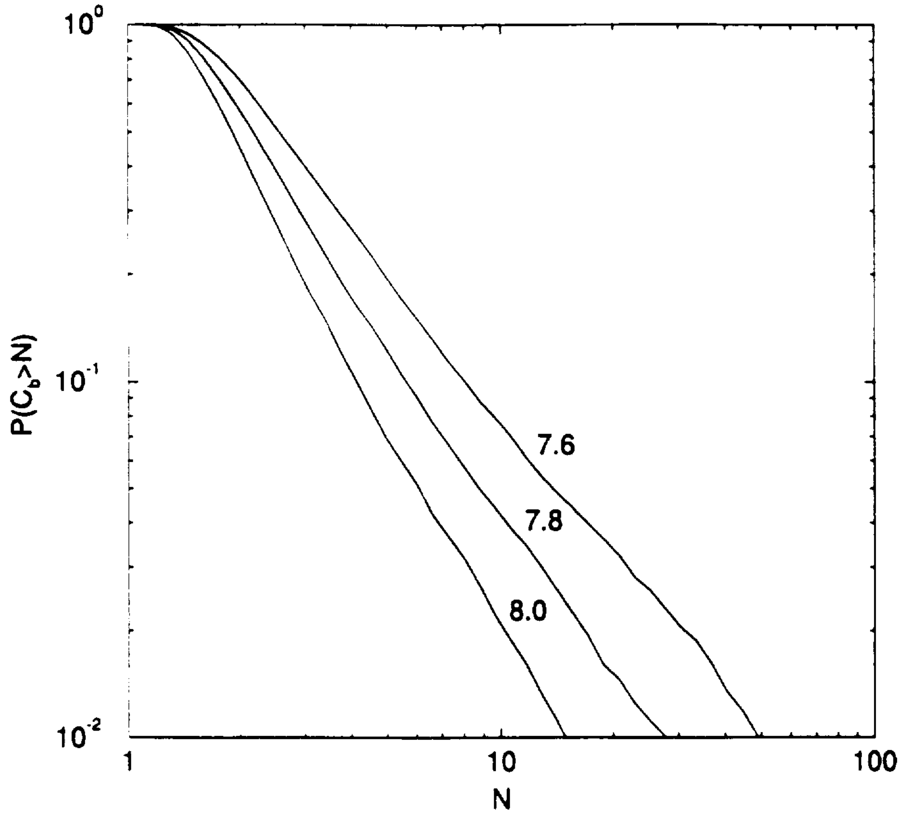


Figure 2.10: Computational distribution for sequential decoding of trellis codes

the computational distribution $Pr(C_b > N)$ of an ODP trellis code with constraint length $\nu = 9$ and block length $L = 256$ at $SNR's$ = 7.6, 7.8, and 8.0 dB, respectively. $Pr(C_b > N)$ is computed using the formula

$$Pr(C_b > N) = \frac{N_C}{N_F}, \tag{2.24}$$

where N_C is the number of blocks for which the number of computations exceeded $N(L+\nu)$ and N_F is the total number of blocks decoded. Each forward look in the Fano Algorithm was counted as one computation. We can see from Figure 2.10 that the computational distribution for sequential decoding of trellis codes can be very well approximated by the Pareto distribution.

3

ERASUREFREE SEQUENTIAL DECODING

In Chapter 2, application of sequential decoding to trellis codes has been investigated. It shows that significant coding gains can be achieved using sequential decoding. However, we note that the computational effort of sequential decoding is a random variable with a Pareto distribution just as in the case of convolutional codes [37, 67]. Thus, although the undetected error probability can be made arbitrarily small, some data cannot be completely decoded and the probability of incomplete decoding is usually on the order of 10^{-2} to 10^{-3} [43]. The performance of sequential decoding is then limited in the case where a feedback channel is not available. However, if the drawback of erasures can be overcome, sequential decoding may be a good alternative to the Viterbi algorithm even if a feedback channel is not available.

Forney and Bower[25] used a simple resynchronization mechanism to avoid the buffer overflow problem in their hardware sequential decoder. Wang and Costello [78, 79] recently have also introduced several erasurefree sequential decoding algorithms. In this chapter, a new erasurefree sequential decoding algorithm is introduced and its application to trellis codes is investigated. In Section 3.1, a general erasurefree sequential algorithm is presented. In Section 3.2, the performance of a block decoding scheme, which guarantees erasurefree decoding, is analyzed. Upper and lower bounds on the bit error rate of the scheme are derived. The performance of erasurefree sequential decoding is compared with the VA by simulation. In Section 3.3, a general resynchronization scheme for continuous sequential decoding is presented. In Section 3.4, the performance of a continuous decoding scheme is studied via simulation and compared with the block decoding scheme and with the VA.

3.1 Erasurefree Decoding - the Buffer Looking Algorithm (BLA)

It has been shown that the computational effort for sequential decoding of trellis codes is a random variable with a Pareto distribution. Thus, the buffer in a sequential decoder will occasionally overflow. This will result in data loss (erasures). On channels where feedback is available, if the buffer overflows, the current block of data can be declared unreliable and a retransmission can be requested [12]. However, this approach cannot be used if a feedback channel is not available. In this section, we present a general erasurefree sequential decoding scheme called the Buffer Looking Algorithm (BLA). The BLA, which includes the algorithms presented in [25], [78], and [79] as special cases, guarantees that the buffer will never overflow and thus that no data will be lost in the process of decoding.

The BLA can be operated in either a block or continuous decoding mode. We will describe them separately, but the general concept of the BLA is illustrated in Figure 3.1. A buffer of size B is divided into M sections, each with size B_j $(1 \leq j \leq M)$. A suboptimum but fast decoding algorithm which can be used as part of the BLA must be selected. One possibility is the M-algorithm[45], which provides a good tradeoff between decoding speed and performance. For systematic codes, the direct recovery of the information bits by making hard decisions on the received sequence

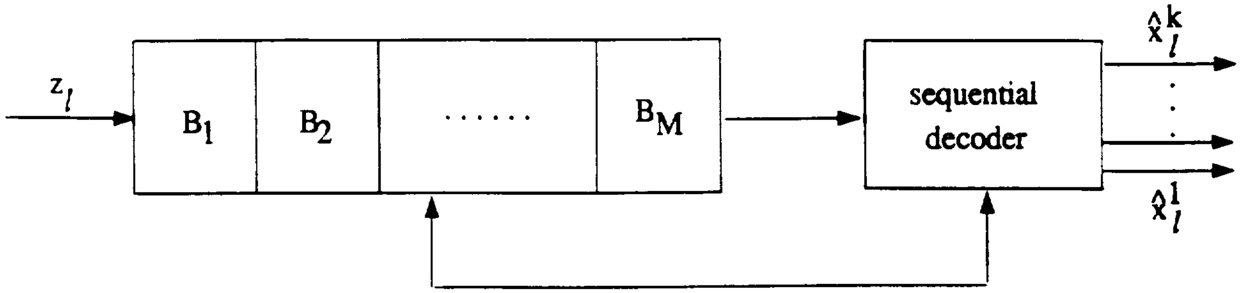


Figure 3.1: Block diagram of the Buffer Looking Algorithm

[25] may be considered. Another possibility is to change the bias k of the Fano metric in conventional sequential decoding[78]. These suboptimum decoding algorithms will be called secondary decoders. The idea of the BLA is to monitor the state of the buffer and to use this information to choose a faster algorithm as the buffer nears saturation. A primary (conventional) sequential decoder and $M-1$ secondary decoders are used. Assume that the j-th secondary decoder is faster but has poorer performance than the i-th secondary decoder if $i < j$. The primary decoder is used when only buffer section B_1 is occupied. while the j-th secondary decoder is used when the first j+1 buffer sections are occupied. The decoder has a core memory that can hold N branch signals and some other necessary information. We let $N = L + \nu$ for the block decoding mode and $N = L_t + 1$ for the continuous decoding mode, where L is the block length, ν is the code constraint length, and L_t is the backsearch limit of a continuous decoder.

First, we describe the BLA in the block decoding mode. In this case, the information sequence is divided into blocks, each with L information branches. After every L information branches, we insert ν all zero branches and then start encoding the next block of L information branches from the all zero state. In this case, we know from which state to start decoding each block of L information branches, and the decoder automatically resynchronizes.

The speed factor, μ, of a sequential decoder is defined as the number of computations that the decoder can perform during the time required to receive one branch (a modulation time period T). Suppose that the decoding speed of the $(M-1)$-th secondary decoder is a constant C_M (computations/branch). Then, the algorithm presented below ·uarantees that the buffer will never overflow if $C_M < \mu$ and $B_M > (L+\nu) \times C_M/\mu$. [he Buffer Looking Algorithm in the Block Decoding mode (BLA-BD) is described as follows.

0) Obtain a block of $L+\nu$ signals from the buffer.

1) If only one buffer section is occupied, decode using the conventional Fano algorithm.

2) If j $(1 < j < M)$ buffer sections are occupied, decode using the (j-1)-th secondary decoder.

3) If all M buffer sections are occupied, go to 6); otherwise, go to 4).

4) If a terminal ($(L+\nu)$-th) node is reached, go to 5); otherwise, go to 1).

5) Release the decoded information bits, obtain the next block of $L+\nu$ signals from the buffer, and reset the decoder to the all zero state. Go to 1).

6) Jump to the best node (the one with the best metric) visited in the block and decode the remaining signals using the (M-1)-th secondary decoder. Release the L decoded branches, obtain the next block of $L+\nu$ signals from the buffer, and reset the decoder to the all zero state. Go to 1).

A flowchart of the BLA-BD is shown in Figure 3.2, where j denotes the number of occupied buffer sections, l denotes the current level of the decoder, and l_b denotes the level of the best node in step 6). In a practical implementation, an interrupt procedure

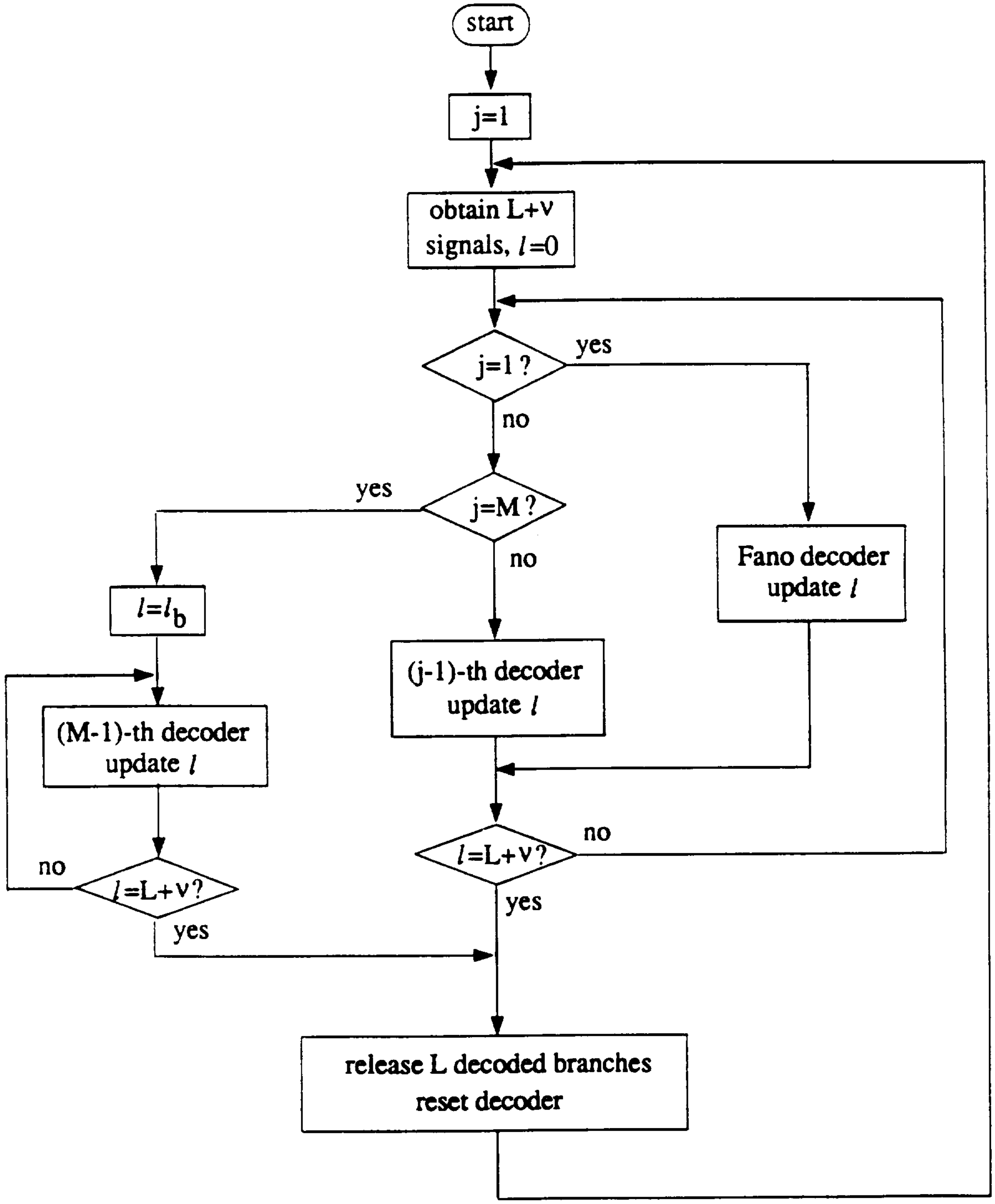

start
j=1
obtain L+ν signals, l=0
j=1 ?
yes
no
j=M ?
yes
no
Fano decoder update l
l=l_b
(j-1)-th decoder update l
(M-1)-th decoder update l
no
l=L+ν?
yes
no
l=L+ν?
yes
release L decoded branches reset decoder

Figure 3.2: A Flowchart of the BLA-BD

can be initiated whenever a new buffer section is occupied. The procedure can be programmed to select the corresponding secondary decoder. A variety of erasurefree algorithms can be obtained from the BLA-BD by using different secondary decoders.

The BLA-BD guarantees resynchronization at the beginning of each block, but it results in some rate loss. For a rate $k/k+1$ trellis code with constraint length ν, the effective informa*t*ion rate is given by

$$R' = \frac{kL}{(k+1)(L+\nu)}, \tag{3.1}$$

when a tail of length ν is transmitted. Large L makes the rate loss small. However. it will be shown that good performance requires relatively small L. This can be viewed as a penalty that must be paid by a block decoder in order to guarantee resynchronization.

Continuous decoding is possible if a resynchronization scheme is available which is capable of recovering from an incorrect path. The resynchronization scheme is used when the decoder gets onto a wrong path and cannot recover on its own. In Section 3.3, a resynchronization scheme is presented that uses hard decisions on received signals. It tries to find one constraint length of error free received signals and use them to resynchronize. If the time required to test one branch in the resynchronization scheme is less than the modulation time period T, the algorithm presented below guarantees that the buffer will never overflow if $B_M > 1$. The algorithm uses a backsearch limit L_t that must be chosen at least four or five times the code constraint length to maintain good performance. The B''ffer Looking Algorithm in the Continuous Decoding mode (BLA-CD) is described as follows.

0) Obtain a block of $L_t + 1$ signals from the buffer.

1) If only one buffer section is occupied, decode using the conventional Fano algorithm.

2) If j ($1 < j < M$) buffer sections are occupied. decode using the (j-1)-th secondary decoder.

3) If all M buffer sections are occupied, go to 6); otherwise, go to 4).

4) If the ($L_t + 1$)-th node is reached, go to 5); otherwise, go to 1).

5) Shift the signals in the core memory one branch (a branch is decoded and released) and obtain the next signal from the buffer. Go to 1).

6) Jump to the deepest node visited by the decoder. Release the branches in the core memory leading to this node (as decoded branches) and obtain the corresponding number of signals from the buffer. Initiate the resynchronization procedure and use the hard decision received signals as decoded branches during the process of resynchronization. When the resynchronization procedure stops, go to 1).

A flowchart of the BLA-CD is shown in Figure 3.3, where j denotes the number of occupied buffer sections, l denotes the current level of the decoder, and l_d denotes the level of the deepest node in step 6). In the BLA-CD, it is impossible for the decoder to move back to the first node in the core memory and to look back from there. In this case, the threshold is lowered to force it to look forward. This automatically imposes a backsearch limit. It is clear from the algorithm that the maximum number of levels that the decoder can look back, i.e., the backsearch limit, is L_t. We will analyze and simulate the performance of the BLA-BD in the next section. The performance of the BLA-CD is studied in Section 3.4. Our results show that the BLA-CD performs essentially the same as the BLA-BD.

Before concluding this section, we observe that the BLA-CD does not suffer from rate loss like the BLA-BD. But since resynchronization schemes are basically probabilistic, it may take a long sequence of received signals to resynchronize successfully. This problem can be alleviated by using a mixed resynchronization scheme. In this case, the data are encoded into blocks with a large block length to minimize the rate

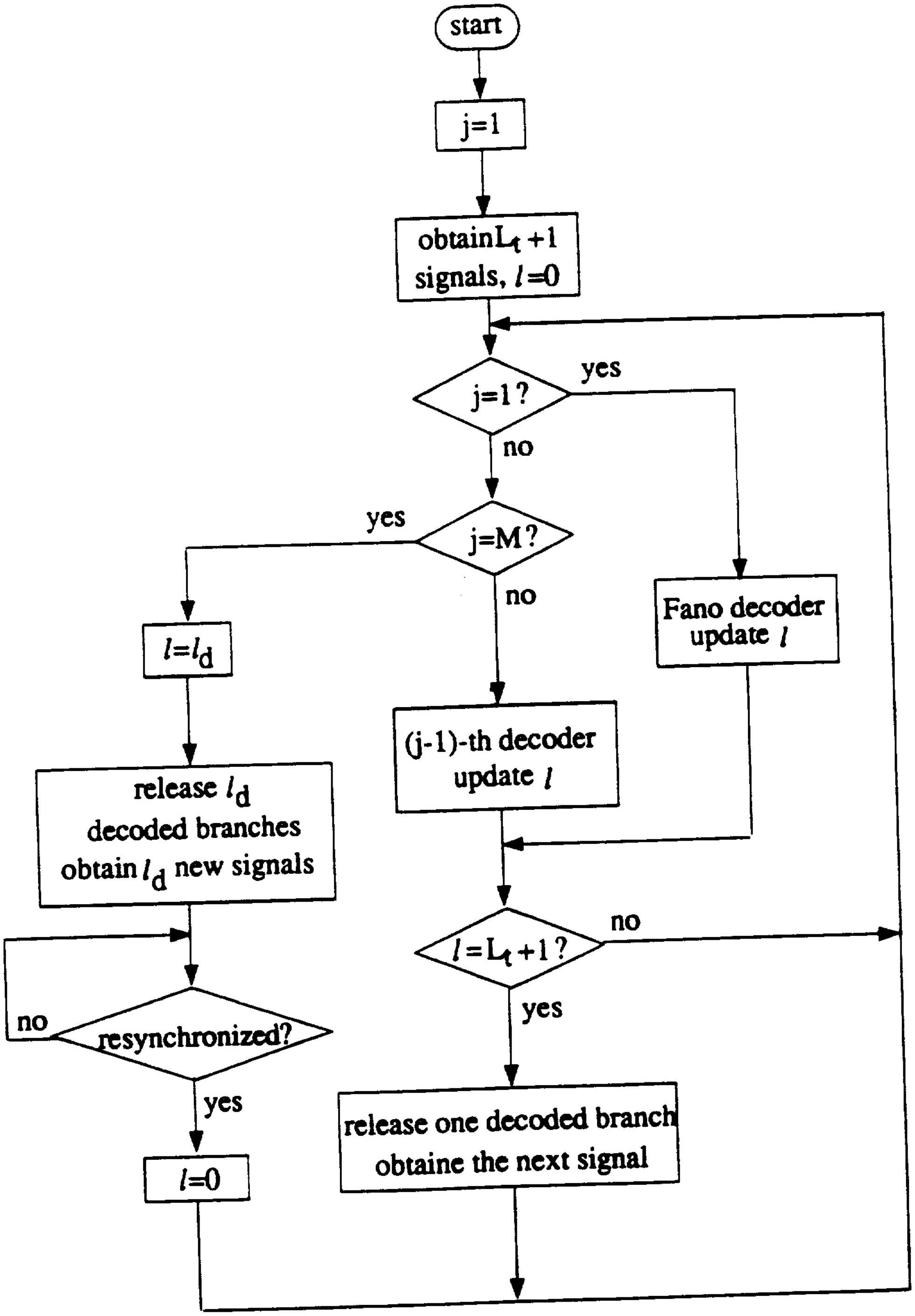

start
j=1
obtain L_t +1 signals, l=0
j=1 ?
yes
no
j=M ?
yes
no
Fano decoder update l
l=l_d
(j-1)-th decoder update l
release l_d decoded branches obtain l_d new signals
l=L_t+1 ?
no
yes
no
resynchronized?
yes
l=0
release one decoded branch obtaine the next signal

Figure 3.3: A Flowchart of the BLA-CD

loss. Then the decoder tries a resynchronization scheme if it loses the correct path. If too long a sequence of received signals is needed to resynchronize, the decoder can resynchronize at the beginning of the next block.

3.2 Performance of the BLA in a Block Decoding Mode

In this section, the performance of a simple BLA-BD scheme is analyzed to determine how the BER of the BLA is affected by parameters such as the speed factor μ and the buffer size B. The same approach can be used to analyze more complex versions of the BLA.

This version of the BLA-BD divides the buffer into two sections and hard decisions are used as estimates of the transmitted information bits by the secondary decoder. This is possible for trellis codes which are constructed in systematic form. In this version of the BLA-BD, step 6) is modified as follows:

6)′ Jump to the best node (the one with the best metric) visited in the block and recover the remaining branches of the block using hard decisions. Obtain the next block of $L + \nu$ signals from the buffer and reset the decoder to the all-zero state. Go to 1).

Also note that, since there are only two sections in the buffer, there is no step 2). The secondary decoder in this version of the BLA-BD is very fast since only one computation is required to decode one branch ($C_M = C_2 = 1$). In this case, the size of the second buffer section B_2 only needs to be larger than $(L + \nu)/\mu$ to guarantee that the buffer will never overflow.

Let P_b be the overall BER of the BLA-BD. The BLA-BD may operate in either one of the two modes, i.e., conventional sequential decoding and suboptimum decoding.

Let P_b^c and P_b^s denote the bit error rate of conventional sequential decoding and suboptimum decoding, respectively. P_b^c and P_b^s can be estimated by simulation or by bounds. Let P_c be the probability that a received branch is decoded by conventional sequential decoding and P_s be the probability that a received branch is decoded by suboptimum decoding. Since $P_c + P_s = 1$, we have

$$\begin{aligned} P_b &= P_c P_b^c + P_s P_b^s \\ &= (1 - P_s) P_b^c + P_s P_b^s. \end{aligned} \tag{3.2}$$

The i-th branch in a block will be decoded by the suboptimum decoder if and only if buffer section B_2 is occupied during the decoding of the block and the i-th branch is beyond the best node (the i_b-th branch), i.e., $i > i_b$. Suppose that the number of computations required to decode one branch is C_b for conventional sequential decoding and that the decoder starts decoding the block with b unoccupied spaces in the buffer. Assume that $B_2 = (L + \nu)/\mu$. Then, the buffer section B_2 will be occupied if $C_b(L+\nu) > (b-(L+\nu)/\mu)\mu$, i.e., $C_b > b\mu/(L+\nu)-1$. The probability that the i-th branch is decoded by the suboptimum decoder given that the decoder starts decoding with b unoccupied spaces in the buffer is then given by

$$\begin{aligned} P(s|b) &= P(i > i_b) \times P\left(C_b > \frac{b\mu}{L+\nu} - 1\right) \\ &= \frac{1}{2} A \left(\frac{b\mu}{L+\nu} - 1\right)^{-\rho}, \end{aligned} \tag{3.3}$$

where $P(C_b > N)$ is the computational distribution of sequential decoding as given in (2.23), A and ρ are constants, and $P(i > i_b)$ denotes the probability that the i-th branch is beyond the best node. ($P(i > i_b)$ is approximated as $1/2$ by assuming that the best node is uniformly distributed in the block[1].) It is easy to see that

[1] This approximation is justified by noting that the best node in a block typically precedes one

$$
\begin{aligned}
P_s &= \int_{L+\nu}^{B} P(b)P(s|b)db \\
&= \int_{L+\nu}^{B} \frac{1}{2}A\left(\frac{b\mu}{L+\nu}-1\right)^{-\rho} P(b)db.
\end{aligned} \tag{3.4}
$$

Note that b is a random variable which ranges from $L+\nu$ to B $(B > L+\nu)$ and primarily depends on μ. We consider three cases. If μ is very small, B_2 will always be occupied. In this case, the buffer will only have $L+\nu$ unoccupied spaces after the previous block is decoded. The distribution of b can then be approximated by

$$P(b) = \delta[b-(L+\nu)], \tag{3.5}$$

where $\delta(\cdot)$ denotes the unit impulse function. Using (3.4) and (3.5), we obtain an upper bound on P_s given by

$$P_s \leq \frac{A}{2(\mu-1)^{\rho}}. \tag{3.6}$$

If μ is very large, the buffer will always be empty. In this case, we may approximate the distribution of b by

$$P(b) = \delta(b-B). \tag{3.7}$$

Using (3.4) and (3.7), we obtain a lower bound on P_s given by

$$P_s \geq \frac{A}{2}\left(\frac{L+\nu}{B\mu}\right)^{\rho}. \tag{3.8}$$

For values of the speed factor μ between the extremes, we assume that b is uniformly distributed over $[L+\nu, B]$, i.e.,

or more noisy branches and that the noise affects the branches of a block independently.

$$P(b) = \begin{cases} \frac{1}{B-(L+\nu)}, & \text{if } b \in [L+\nu, B] \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

In this case, we have

$$\begin{aligned} P_s &= \int_{L+\nu}^{B} \frac{1}{B-(L+\nu)} \times \frac{1}{2} A \left(\frac{b\mu}{L+\nu} - 1 \right)^{-\rho} db \\ &= \frac{1}{2} \times \frac{A}{B-(L+\nu)} \times \left(\frac{L+\nu}{\mu} \right)^{\rho} \int_{L+\nu}^{B} \left(b - \frac{L+\nu}{\mu} \right)^{-\rho} db \\ &= \frac{A}{2(\rho-1)} \times \frac{1}{B-(L+\nu)} \times \left(\frac{L+\nu}{\mu} \right)^{\rho} \\ &\quad \times \left\{ \left[\frac{(L+\nu)(\mu-1)}{\mu} \right]^{1-\rho} - \left(B - \frac{L+\nu}{\mu} \right)^{1-\rho} \right\}. \end{aligned} \tag{3.10}$$

In the case when $B \gg L+\nu$, the second term in the brackets is much smaller than the first one since $\rho > 1$. Thus, P_s can be approximated by

$$P_s \approx \frac{A}{2(\rho-1)} \times \frac{L+\nu}{B\mu(\mu-1)^{\rho-1}}. \tag{3.11}$$

The overall approximate BER and its lower and upper bounds can be obtained by substituting the approximate P_s and its lower and upper bounds into (3.2). Substituting (3.11) into (3.2), we obtain

$$\begin{aligned} P_b &\approx \left(1 - \frac{A}{2(\rho-1)} \times \frac{L+\nu}{B\mu(\mu-1)^{\rho-1}} \right) P_b^c \\ &\quad + \frac{A}{2(\rho-1)} \times \frac{L+\nu}{B\mu(\mu-1)^{\rho-1}} P_b^s \end{aligned} \tag{3.12}$$

In the normal operating range of the decoder, $P_s \ll 1$. Then, we have

$$P_b \approx P_b^c + \frac{A}{2(\rho-1)} \times \frac{L+\nu}{B\mu(\mu-1)^{\rho-1}} P_b^s. \tag{3.13}$$

Similarly, we can obtain upper and lower bounds on P_b by applying (3.6) and (3.8) to (3.2) as follows.

$$P_b^c + \frac{A}{2}\left(\frac{L+\nu}{B\mu}\right)^{\rho} P_b^s \leq P_b \leq P_b^c + \frac{A}{2(\mu-1)^{\rho}} P_b^s. \tag{3.14}$$

In the case when $P_b^c \ll P_b^s$, P_b is primarily determined by P_s, which is a function of the speed factor μ, the buffer size B, the block length L, the constraint length ν, and the parameters A and ρ. ν and ρ ($\rho > 1$ if the code rate is smaller than the channel cut-off rate) are determined by the code and the channel SNR, respectively, while A is typically between 1 and 10 depending on the particular version of sequential decoding employed [25, 31, 64]. Thus, μ, B, and L are the parameters of the BLA-BD that determine its performance. The best compromise for the BLA-BD is to choose μ, B, and L such that the two terms in the P_b expression are comparable or the second term is smaller. (3.13) and (3.14) show that it generally requires large B, large μ, and small L to reduce the BER. In the rest of this section, we present simulation results that verify our analysis. In the simulations, ODP trellis coded 8-PSK with $\nu = 9$ was used.

(3.13) and (3.14) show that μ is the most critical parameter determining the BER of the BLA-BD. Figure 3.4 shows the overall BER P_b as a function of μ with $B = 16$ K symbols. It shows that the BER decreases rapidly with increasing μ. When μ becomes greater than about 5, the number of errors contributed by the suboptimum decoder becomes negligible and the BER decreases very little with further increases in μ.

The upper bound (the small μ case) indicates that the BER is independent of the buffer size. Figure 3.5 shows the BER of the BLA-BD with $L = 128$ symbols, $\mu = 3$, and $SNR = 7.6$ dB as a function of the buffer size. $\mu = 3$ is smaller than the average number of computations for sequential decoding at $SNR = 7.6$ dB. Thus, the curve is quite flat, agreeing with our analysis. For a moderate speed factor μ, the

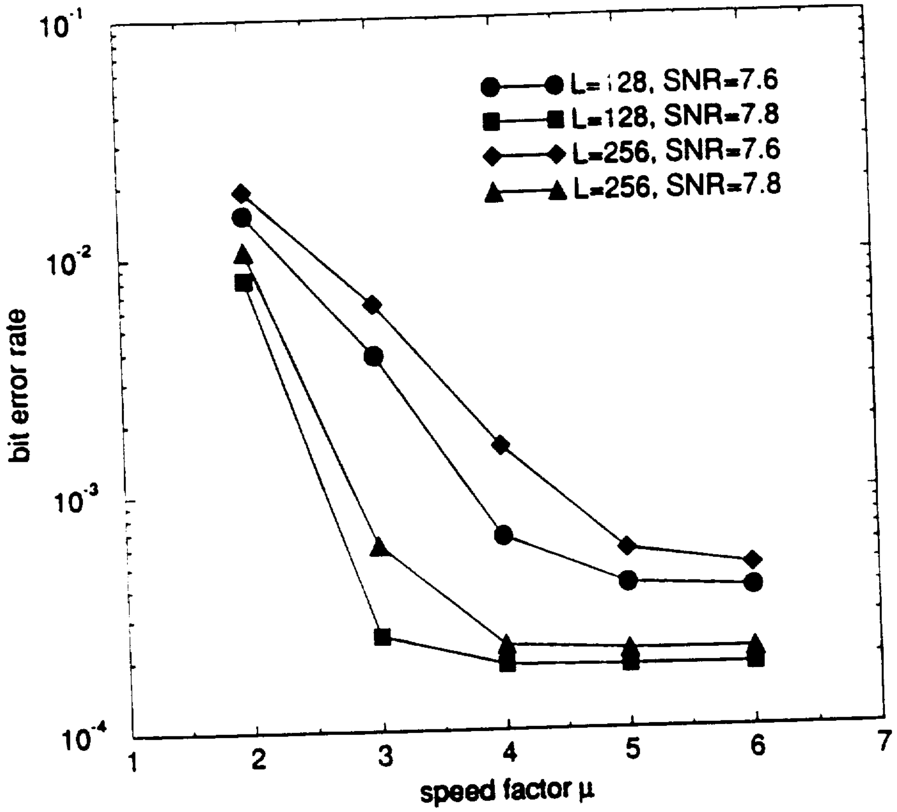


Figure 3.4: Influence of speed factor on the bit error rate

BER is expected to decrease with increasing buffer size until the number of errors contributed by the suboptimum decoder becomes negligible compared with P_b^c. The other four curves in Figure 3.5 show simulation results with L = 128 symbols, $\mu = 3$, and $SNR = 7.8$ dB; L = 128 symbols, $\mu = 4$, and $SNR = 7.8$ dB; L = 256 symbols, $\mu = 3$, and $SNR = 8.0$ dB; and L = 256 symbols, $\mu = 4$, and $SNR = 8.0$ dB, respectively. These curves clearly illustrate the expected behavior.

As shown in Figure 2.7 and discussed in Section 2.5, the BER is not a function of the block length for conventional complete sequential decoding as long as a one constraint length tail is added. However, (3.13) and (3.14) show that the BER of the BLA-BD does depend on the block length. Figure 3.6 shows the BER as a function of

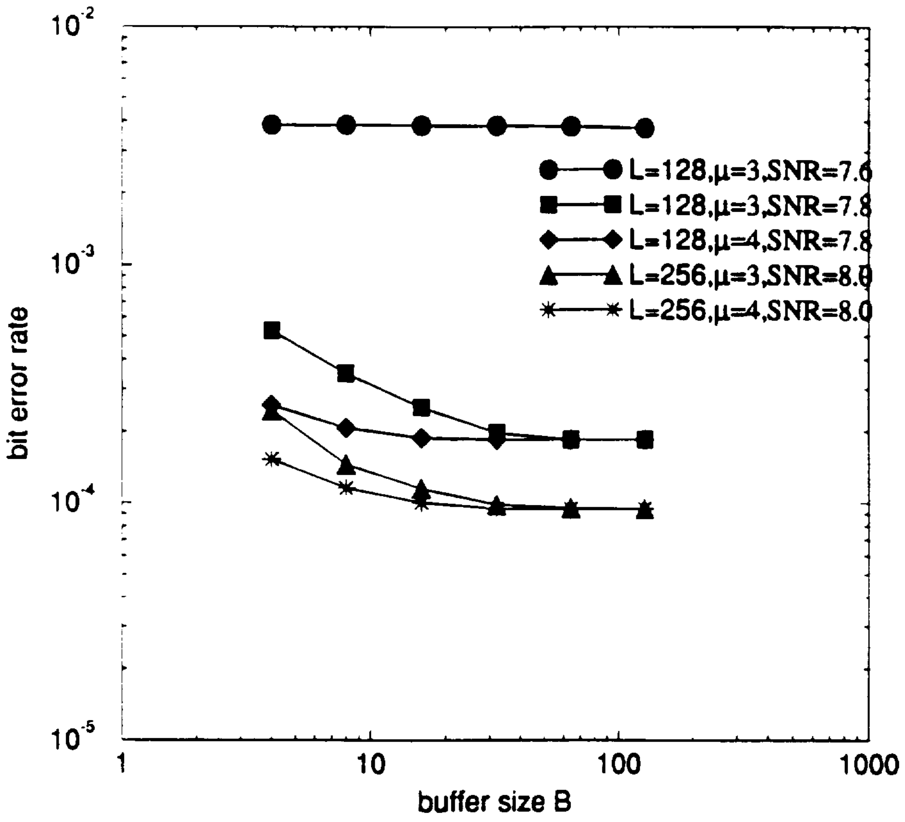


Figure 3.5: Influence of buffer size on the bit error rate

L with $B = 16$ K symbols. Intuitively, for smaller block lengths, less data is decoded by the suboptimum decoder and thus the BER is smaller. The simulation results and the analysis are both consistent with this intuition. (On the other hand, smaller blocks result in more rate loss as shown in (3.1) and Figure 3.6 does not take the rate loss into account.)

The above analysis and simulation results show that it requires large μ, large B, and small L to achieve a small BER. In practice, we may first choose L such that the rate loss is tolerable and then choose B and μ such that the number of errors contributed by the suboptimum decoder becomes negligible compared to P_b^c.

Figure 3.7 shows the performance of the VA with constraint length $\nu = 6$ Unger-

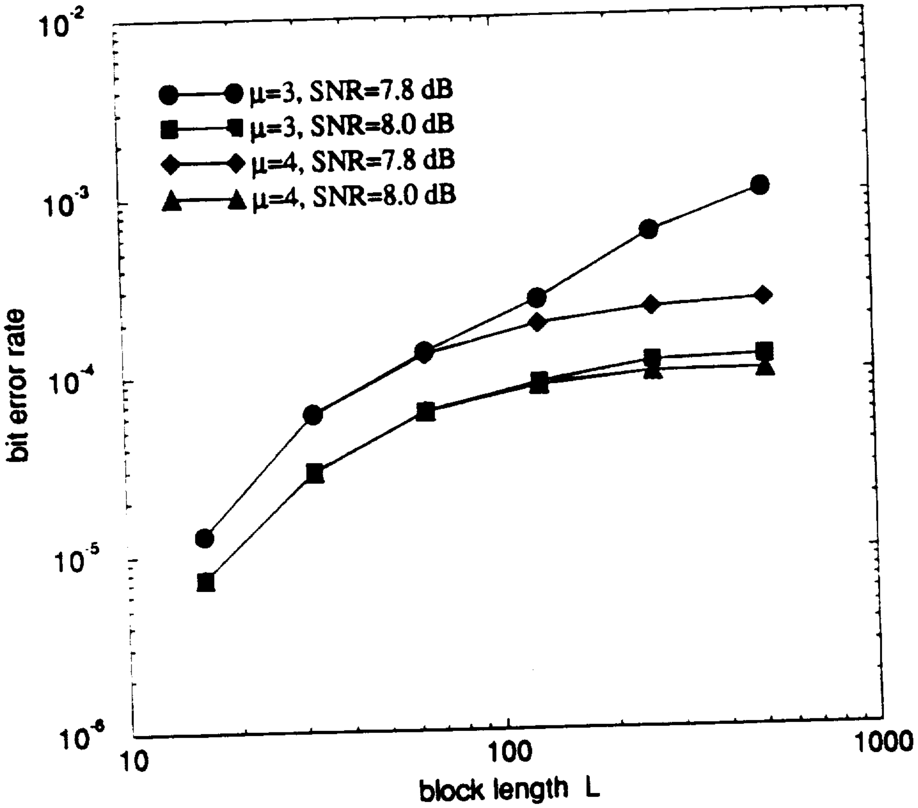


Figure 3.6: Influence of block length on the bit error rate

boeck trellis coded 8-PSK and the cut-off rate bound for 8-PSK modulation at a spectral efficiency of 2 bits/T. The other two curves in Figure 3.7 show the performance of the BLA-BD with constraint length $\nu = 10$ ODP trellis coded 8-PSK [85], buffer size $B = 32$ K symbols, speed factor $\mu = 4$, and block length $L = 256$ symbols; and constraint length $\nu = 13$ ODP trellis coded 8-PSK [85], $B = 32$ K symbols, $\mu = 6$, and $L = 256$ symbols, respectively. These results show that the BLA-BD with $\nu = 13$ can achieve more than 1.0 dB of coding gain over the VA and is only about 0.3 dB away from the cut-off rate bound at a BER of 10^{-5}.

A typical computation in sequential decoding involves regenerating code branches, finding the branch metrics, computing the path metrics, and choosing the path with

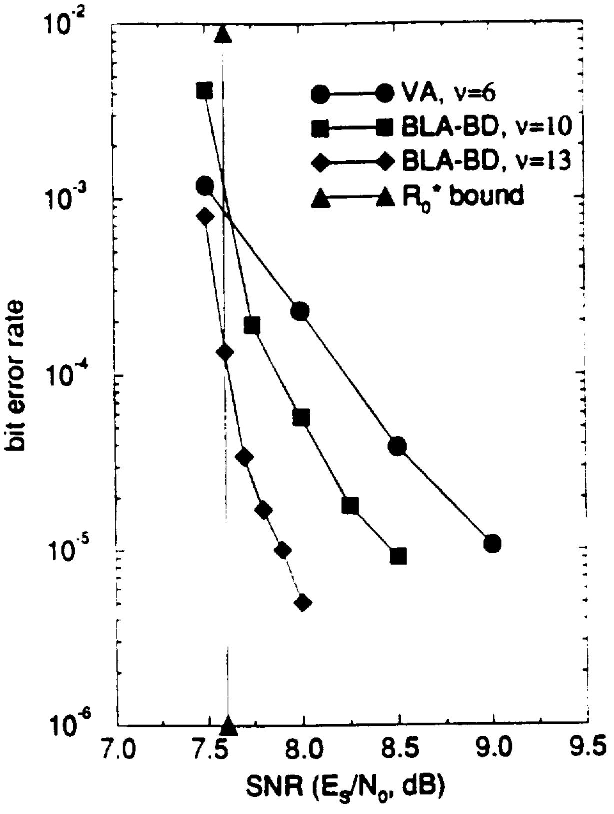


Figure 3.7: Performance of the BLA-BD

the best metric. These operations are also needed by a Viterbi decoder at each state in the code trellis. Thus, a computation in the BLA-BD is comparable to a computation in the VA. The speed factor μ for sequential decoding is comparable to the number of states 2^{ν} in the VA since μ is the maximum average number of computations which a sequential decoder is allowed for decoding one branch, whereas the VA requires 2^{ν} computations per branch. For the codes in Figure 3.7, the VA requires 64 computations per branch since the $\nu = 6$ code has 64 states, whereas the BLA-BD with $\nu = 13$ requires an average of at most 6 computations per branch. Thus, the superior performance of the BLA-BD over the VA is achieved with much less computational effort. Note that the BLA-BD curve with $\nu = 13$ shown in Figure

3.7 loses about 0.2 dB at a BER of 10^{-5} compared with Curve SD, $\nu = 13$, in Figure 2.9. However, the simulation results in Figure 2.9 were obtained using a complete decoder that r·quires an infinite buffer while the BLA-BD uses a finite buffer. This implies that ·· modest loss in coding gain is the price that must be paid for practical sequential decoding.

3.3 The Problem of Resynchronization in Continuous Sequential Decoding

It is widely believed that continuous sequential decoding does not have good resynchronization capability and thus block decoding is usually preferred [40, 43, 63, 69]. This results in some rate loss, which is undesirable. However, Forney and Bower[25] have used a backsearch limited Fano Algorithm[14] i.. onjunction with a simple resynchronization mechanism in a hardware implementation of a continuous sequential decoder using a systematic feedforw·:d constraint length $\nu = 47$ convolutional code. It is easy to see that resynchronization will be successful for a systematic feedforward convolutional code if one constraint length of correctly received data is fed into the encoder replica at the receiver (which will be called the recoder following the terminology of Forney and Bower[25]). (Another advantage of using systematic codes is that the information bits can easily be estimated directly from the received sequence during the process of resynchronization.) But the resynchronization of a sequential decoder for more powerful non-systematic feedforward codes and systematic feedback codes remains a problem. In this section, we introduce a general resynchronization scheme which allows a sequential decoder to be resynchronized for non-systematic feedforward and systematic feedback codes.

First, we note that it is always possible to convert a non-systemat·· feedforward encoder to an equivalent systematic feedback encoder[16, 60]. Thus, we only need to devise a resynchronization scheme for systematic feedback codes. For simplicity, we only consider rate $k/k+1$ codes, but the scheme can easily be generalized to codes with other rates. A general implementation of a rate $k/k+1$ systematic feedback recoder is shown in Figure 3.8 (switch S is closed), where h_j^i ($i = 0, 1, \cdots, k$ and $j = 0, 1, \cdots, \nu$)

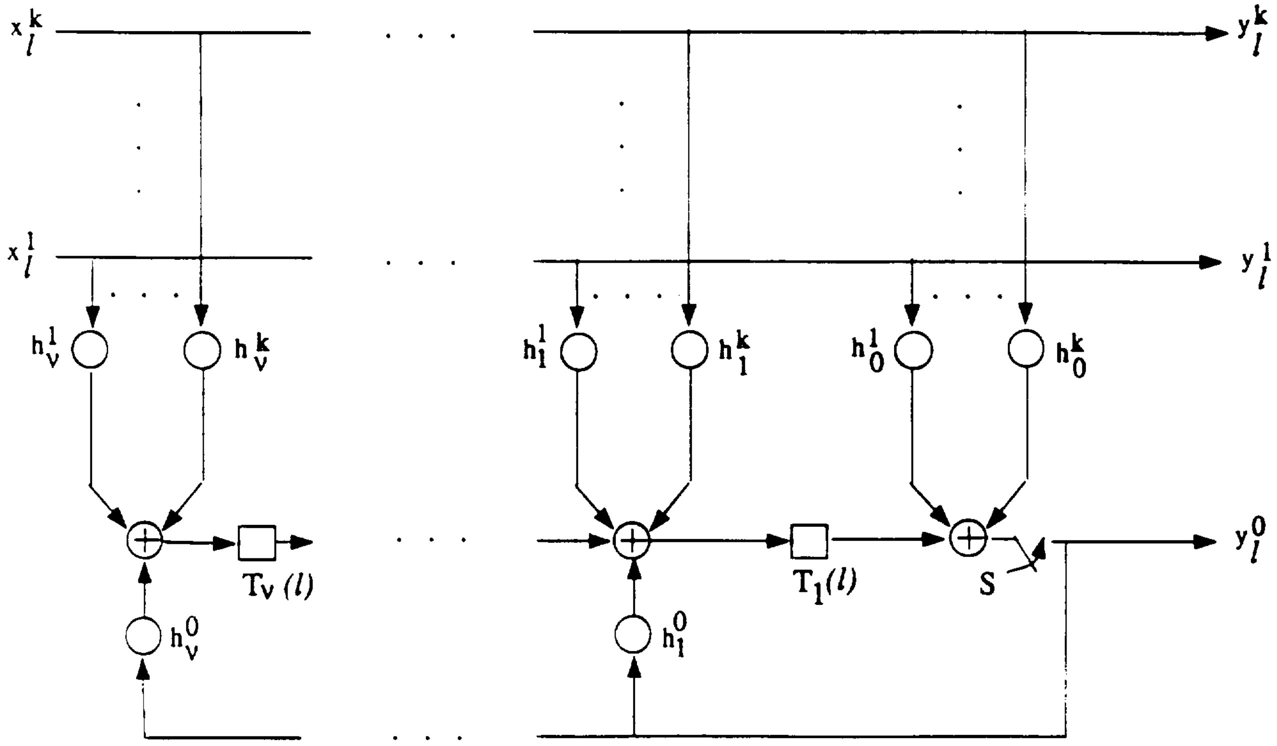


Figure 3.8: The implementation of a systematic feedback recoder

denotes the code parity-check coefficients, $\mathbf{T}(l) = [T_1(l), T_2(l), \cdots, T_\nu(l)]$ denotes the recoder state at time unit l, and $\mathbf{x}_l = [x_l^1, \cdots, x_l^k]$ and $\mathbf{y}_l = [y_l^0, y_l^1, \cdots, y_l^k]$ denote the recoder input and output vectors at time unit l, respectively. In normal operation, $\mathbf{x}_l$ is a hypothetical information input and $\mathbf{y}_l$ is the corresponding output. However, during the process of resynchronization, we assume that $\mathbf{y}_l$ is obtained directly from the channel (for convolutional codes) or by making hard decisions (for trellis codes) and that $y_l^0, y_l^1, \cdots, y_l^k$ are used as the inputs of the recoder. Resynchronization is the process of finding the correct state of the recoder from an incorrect state.

A sequential decoder moves forward and backward in the code tree. In practice, however, the decoder will not be allowed to move more than some maximum number of levels back from its deepest penetration into the code tree. This is similar to path truncation in the Viterbi algorithm[73]. If errors occur in a decoded sequence, the correct path will be lost and the recoder will enter an incorrect state. In this case, it is impossible for the recoder to get into the correct state again unless certain error patterns occur. The following recursive equations describe the state transitions in the code trellis. The state of the recoder at time l, $\mathbf{T}(l) = [T_1(l), T_2(l), \cdots, T_\nu(l)]$, is related to its previous state and the current output vector by

$$\begin{aligned} y_l^0 &= T_1(l-1) \oplus \sum_{i=1}^{k} h_0^i y_l^i, \\ T_j(l) &= T_{j+1}(l-1) \oplus \sum_{i=0}^{k} h_j^i y_l^i, \quad 1 \le j \le \nu - 1, \\ T_\nu(l) &= \sum_{i=0}^{k} h_\nu^i y_l^i, \end{aligned} \tag{3.15}$$

where $\oplus$ and $\sum$ both denote modulo-2 addition. From (3.15), it can be seen that the state will normally still be incorrect if the previous state is incorrect. Furthermore, the correct state may never be found because y_l^0 is related to $T_1(l-1)$, which will be incorrect for some l, and all the other state variables are related to y_l^0.

Note that $\mathbf{y}_l$ is known for convolutional codes and can be found by making hard decisions for trellis codes. Now consider turning the switch S off in Figure 3.8 and using $\mathbf{y}_l$ as the input. Since it is only related to $\mathbf{y}_l$, $T_\nu(l)$ will be correct if $\mathbf{y}_l$ is correct and will be given by the following recursive equations,

$$\begin{aligned} y_l^0 &= y_l^0, \\ T_j(l) &= T_{j+1}(l-1) \oplus \sum_{i=0}^{k} h_j^i y_l^i, \quad 1 \leq j \leq \nu - 1, \\ T_\nu(l) &= \sum_{i=0}^{k} h_\nu^i y_l^i. \end{aligned} \tag{3.16}$$

Using the above recursive equations ν times, we will find a correct state if ν consecutive $\mathbf{y}_l$'s are correct. For reference, we will refer to the above straightforward resynchronization method as scheme 1. Let ϵ be the probability that $\mathbf{y}_l$ is incorrect. The probability of successful resynchronization for scheme 1 is given by

$$P_{sr} = (1-\epsilon)^\nu. \tag{3.17}$$

P_{sr} is the probability that one constraint length of received (hard decision) data is error free, since resynchronization is successful if and only if ν consecutive error free data branches are fed into the recoder. The problem is to recognize that ν consecutive branches of data are error free, i.e., to know when to stop the resynchronization process. Note that the cumulative Fano metric is large and decoding is fast if the recoder is in the correct state. Thus, the decoding speed and the Fano metric can be used as measures of when the resynchronization process should be stopped. The following algorithm uses a simpler measure to stop resynchronization. After a resynchronization trial, r branches of received data are decoded and the decoded data are compared with the received data. If all r branches agree, resynchronization is stopped.

A Resynchronization Algorithm for Systematic Feedback Codes:

0) Select $r \geq 1$ and let j be the deepest node visited by the decoder.

1) Set $i = 1$, turn S off, and feed $\mathbf{y}_{j+1}$-$\mathbf{y}_{j+\nu}$ into the recoder shown in Figure 3.8 (see (3.16)).

2) Turn S on and feed $[y^1_{j+\nu+i}, y^2_{j+\nu+i}, \cdots, y^k_{j+\nu+i}]$ into the recoder to obtain $\hat{y}^0_{j+\nu+i}$ (see (3.15)).

3) If $\hat{y}^0_{j+\nu+i} \neq y^0_{j+\nu+i}$, shift the signals in the core memory one branch, use the hard decision on the $(j+1)$-th received signal as the decoded branch, obtain the next signal from the buffer, set $j \leftarrow j+1$, and go to 1). Otherwise, go to 4).

4) If $i < r$, set $i \leftarrow i+1$, and go to 2). Otherwise. release r branches in the core memory (as decoded branches), obtain the corresponding number of signals from the buffer, and go to 5).

5) Stop.

Scheme 1 can be viewed as a special case of this algorithm with $r = 0$. The probability that the algorithm stops within $N \geq \nu + r$ branches of received data is difficult to derive exactly but can be lower bounded by

$$P(N) \geq 1 - \left[1 - (1-\epsilon)^{\nu+r}\right]^{N-\nu-r} . \tag{3.18}$$

$P(N\}$ approaches 1 very quickly with N, since ϵ is normally much smaller than 1.

If the time required to test one branch is less than the modulation time period T, the input buffer will not overflow. From the algorithm. we see that it takes at most r iterations of step 2) to step 4) to test (decode) one branch. The main operation in each iteration is step 2), which is used to compute $\hat{y}^0_{j+\nu+i}$. This requires about $1/2^k$ of the time required to generate the 2^k branches leaving each node in a rate $k/k+1$ code tree. Thus, 2^k iterations of this algorithm are comparable to one computation in a sequential decoder. The algorithm thus guarantee that the input buffer will not overflow if $r \leq 2^k\mu$, where μ is the speed factor of the sequential decoder.

When the algorithm terminates, resynchronization may not be successful. The probability of successful resynchronization for the algorithm is a function of r. We denote this probability as $P_{sr}(r)$. For $r = 0$, $P_{sr}(r) = P_{sr}$ as given in (3.17). However, for $r > 0$, it is difficult to determine $P_{sr}(r)$ analytically since it is related to the code structure. We have used simulations to study the influence of r on $P_{sr}(r)$. Figure 3.9 shows $P_{sr}(r)$ as a function of r for an ODP trellis code with 8-PSK modulation [84, 85] and $\nu = 10$ at an $SNR = 7.6$ and 7.8 dB, respectively. Note that the probability of successful resynchronization achieves a maximum for r around 6. Further increases in r will result in smaller $P_{sr}(r)$ since $(1-\epsilon)^{\nu+r}$, which reflects the probability that consecutive received vectors are error free, declines with increasing r. For smaller r, agreement of r branches of decoded data and received data may not reflect successful resynchronization. It is seen that the maximum probability of successful resynchronization is about 0.4, i.e., two or three trials will normally result

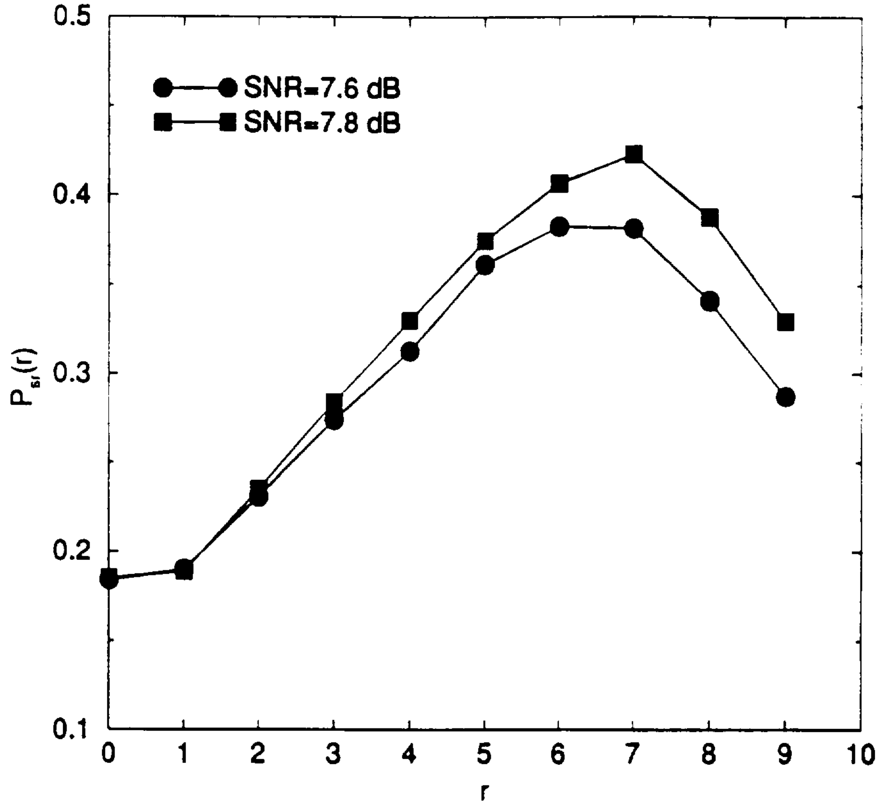


Figure 3.9: Probability of successful resynchronization vs. r

in successful resynchronization.

3.4 Performance of the BLA in a Continuous Decoding Mode

If a continuous sequential decoder cannot resynchronize after losing the correct path, it may flounder forever and give very poor performance. The resynchronization scheme proposed in the previous section has a high probability of successful resynchronization. In this section, we show that significant coding gains are possible with the BLA in a Continuous Decoding mode (BLA-CD) using this resynchronization scheme.

In the BLA-CD, a backsearch limit L_t is imposed. The best value of L_t is determined by trial-and-error. A large L_t requires more memory and can result in excessive searches which cause the buffer to near saturation and initiate a resynchronization process under noisy channel conditions. A small L_t, on the other hand, forces premature threshold lowerings which cause the decoder to accept errors. The choice of L_t thus involves trade-offs between cost and performance. In the following simulations, we have selected L_t so that no significant additional coding gain can be obtained by selecting a larger L_t.

The version of the BLA-CD used in the simulations is similar to the BLA-BD of last section. The buffer is again divided into two sections. The BLA-CD enters a resynchronization mode when the second buffer section is occupied. Figure 3.10 shows the performance of the BLA-CD with constraint length $\nu = 10$ ODP trellis coded 8-PSK [84, 85], $r = 6$, buffer size $B = 32$ K symbols, speed factor $\mu = 4$, and backsearch limit $L_t = 120$ branches; and constraint length $\nu = 13$ ODP trellis coded 8-PSK[84, 85], $r = 6$, $B = 32$ K symbols, $\mu = 6$, and $L_t = 220$ branches.

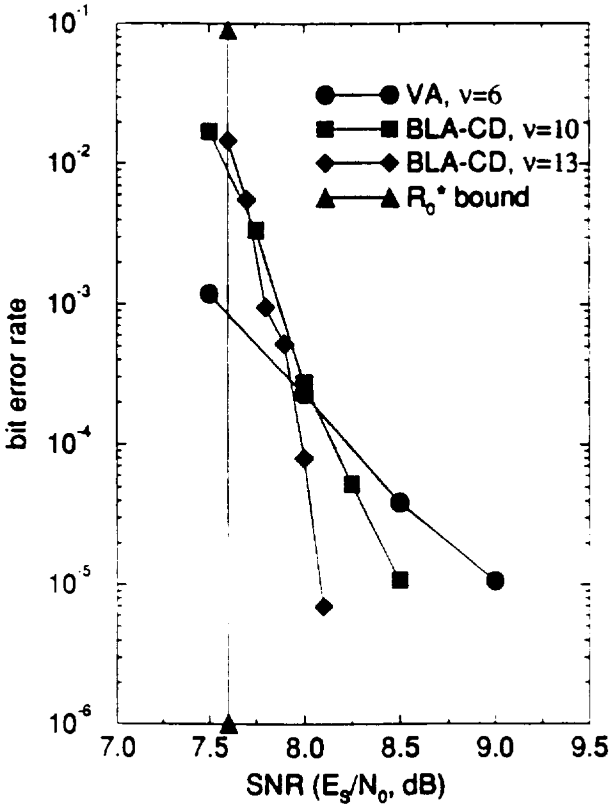


Figure 3.10: Performance of the BLA-CD

For comparison, the performance of the VA with an Ungerboeck $\nu = 6$ code and the cut-off rate bound for 8-PSK modulation at a spectral efficiency of 2 bits/T are also shown. The results show that the resynchronization scheme works quite well and the BLA-CD with $\nu = 13$ can achieve nearly 1.0 dB coding gain over the VA and is only about 0.5 dB away from the cut-off rate bound at a BER of 10^{-5}. (Again note that the BLA-CD with $\nu = 13$ loses about 0.4 dB at a BER of 10^{-5} compared with Curve SD, $\nu = 13$, in Figure 2.9, which was obtained using a complete decoder that requires an infinite buffer.)

The BLA-BD with constraint length $\nu = 13$ code and block length $L = 256$ symbols shown in Figure 3.7 can achieve about 0.2 dB more coding gain over the

BLA-CD with the same code at a BER of 10^{-5}. With these parameters, the rate loss of the BLA-BD caused by adding a 13 branch tail to each block is about 0.2 dB. Thus, the BLA-BD and the BLA-CD are comparable in terms of error performance and energy efficiency at a BER of 10^{-5}. However, the BLA-CD has a slight edge since it maintains a spectral efficiency of 2 bits/T while the effective spectral efficiency of the BLA-BD is about 1.9 bits/T. On the other hand, we can see from Figures 3.7 and 3.10 that the BLA-BD performs better than the BLA-CD at lower SNR's. At higher SNR's the performance gap between the BLA-BD and BLA-CD becomes smaller and their performance is expected to be identical asymptotically.

4

CONSTRUCTION OF ROBUSTLY GOOD TRELLIS CODES

In Chapters 2 and 3, we showed that significant coding gains can be achieved with less computational complexity using sequential decoding and its modifications compared with the Viterbi algorithm. Pottie and Taylor[63] compared the performance of trellis codes using the Viterbi algorithm, the M-algorithm [45], the Fano algorithm[14], and the generalized stack algorithm[32] and similar conclusion was drawn. Thus, sequential decoding appears to be a good alternative to the Viterbi algorithm for trellis codes. However, very few papers[49, 61] have addressed the problem of constructing trellis codes for use with sequential decoding.

Traditionally, the Viterbi algorithm [73] was assumed for decoding trellis codes. The asymptotic error performance of the Viterbi algorithm is determined by the minimum free Euclidean distance of the code[74, 95]. Thus, the free distance has been used as the main criterion in the code construction for use with the Viterbi algorithm [5, 20, 21, 41, 58, 59, 61, 70, 72, 88]. Trellis codes with one and two dimensional constellation were presented by Ungerboeck in [70]. These codes achieve coding gains up to 6.0 dB over uncoded systems with constraint lengths up to 10 . This work was extended for multidimensional constellation in [5, 58, 59, 88] to achieve code rotational invariance among other characteristics. All these codes, which are intended for use with the Viterbi algorithm, are short.

Porath and Aulin[61] constructed large constraint length trellis codes using construction algorithms which extend a subset of subcodes with good distance growths. Their main purpose, however, was still to construct codes with large free distances. In [49], Malladi et. al. attempted to construct trellis codes in systematic feedforward form with good distance profiles which are intended for use with sequential decoding. However, the free distance of their systematic feedforward codes are much smaller than systematic feedback or non-systematic trellis codes for the same constraint length. In this chapter, we construct optimum as well as robustly good large constraint length trellis codes for use with sequential decoding. In Section 4.1, the relationship between the code distance parameters and the computational distribution of sequential decoding is studied. In Section 4.2, trellis codes with Optimum Distance Profiles (ODP) and Optimum Free Distances (OFD) are constructed and the design criterion for trellis codes with sequential decoding is discussed. In Section 4.3, a new approach is proposed to construct robustly good trellis codes. In Section 4.4, simulation results are presented to show that the new codes can perform better than the best known codes when sequential decoding is used.

4.1 Computational Effort of Sequential Decoding

It has been shown [37, 38, 67] that the computational effort of sequential decoding for convolutional codes can be approximated by a Pareto distribution, i.e.,

$$Pr(C_b > N) = AN^{-\rho}, \tag{4.1}$$

where A and ρ are constants. In [37], it is determined that ρ is related to the code rate R by

$$R = \frac{E_0(\rho)}{\rho}, 0 < R < C, \tag{4.2}$$

where C is the channel capacity and $E_0(\rho)$ is the Gallager function. It is assumed that the channel is memoryless with a discrete input and a discrete output in the derivation of (4.1). This assumption can still be regarded valid in the case of bandlimited Additive White Gaussian Noise (AWGN) channel for trellis codes. Thus, the computational distribution of sequential decoding for trellis codes will still be Paretian. This has been verified in Chapter 2 by simulation results. We give one more example to illustrate this. Figure 4.1 shows the computational distribution of sequential decoding with a constraint length $\nu = 8$ trellis code for 8-PSK modulation

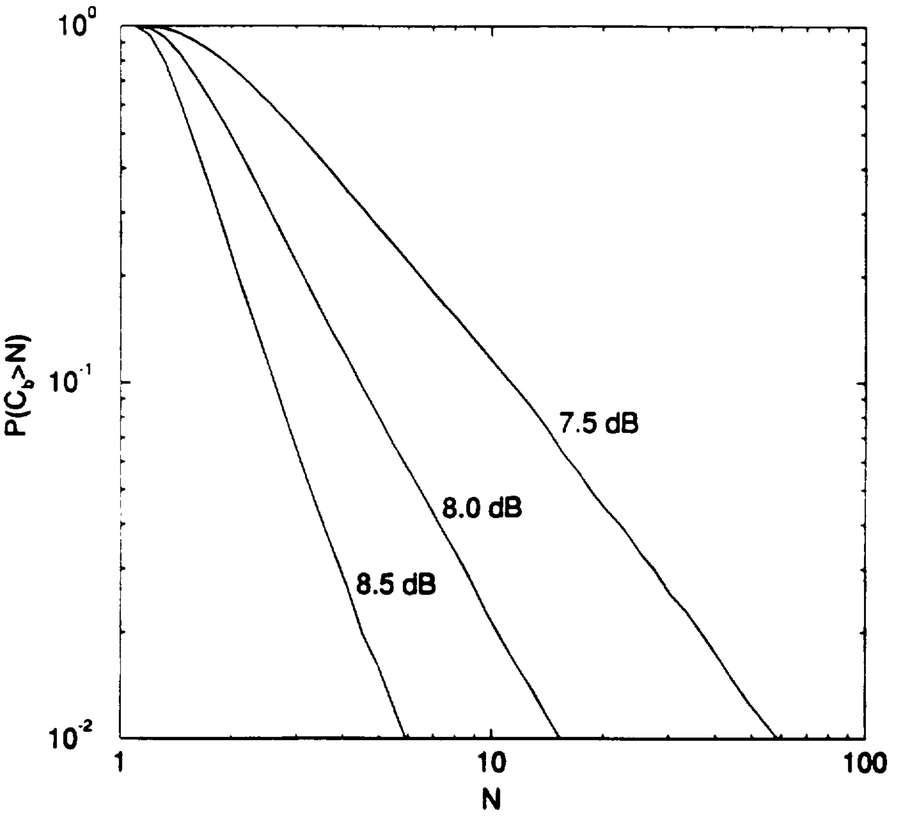


Figure 4.1: Computational distribution for sequential decoding of Ungerboeck code

taken from [70] at $SNR = 7.5, 8.0$, and 8.5 dB, respectively. The same Fano sequential decoder is used and $P_r(C_b > N)$ is defined as in (2.24). It clearly shows that the distributions are very well matched with Pareto distribution.

From (4.1), it is seen that ρ is a critical parameter that determines the moments of computations. (4.2) implies that ρ is related to the code rate which reflects the channel SNR. This is demonstrated in Figure 4.1. It shows that lower code rate (higher SNR) results in larger ρ and thus less computational effort. However, we are more interested in the relationship between ρ (computational effort) and the code structure for a given SNR. Simulations of various trellis codes with a variety of constraint lengths show that ρ is indeed related to specific code structures. In this section, we study how the code structure reflected by the distance parameters influence the computational effort.

Sequential decoding needs to compare the paths of different lengths. We have shown that the Fano metric is an optimum metric for comparison of variable length codes and thus it is used in sequential decoding algorithms. From (2.19), we obtain the branch Fano metric

$$M_B(a_l, z_l) = \log_2 \frac{\exp(-|z_l - a_l)|^2/2\sigma^2)}{\sum_{i=0}^{K-1} \exp(-|z_l - a^i|^2/2\sigma^2)} - (k+1)(1-R), \tag{4.3}$$

where a_l and z_l are the hypothetical channel signal and the received signal at time l, respectively, a^i is the i-th point in the constellation, R is the code rate defined by $R = k/(k+1)$ for a trellis code, and K is the total number of signal points in the modulation constellation. To simplify the discussion, we may rewrite (4.3) as

$$M_B(a_l, z_l) = -\alpha d^2[z_l, a_l] + \beta(z_l), \tag{4.4}$$

where $d^2[z_l, a_l] = |z_l - a_l|^2$ is the Euclidean distance between z_l and a_l, α is a positive constant and $\beta(z_l)$ is a constant independent of the (hypothetical) transmitted signal.

For a partial path of length $l_n + 1$ branches, the cumulative Fano metric is then given by

$$M(l_n) = -\sum_{l=0}^{l_n} \{\alpha d^2[z_l, a_l] + \beta(z_l)\}. \tag{4.5}$$

Assume that the channel is quiet (noise-free). Then z_l will be equal to a_l if the decoder follows the correct path. On the other hand, z_l is not equal to a_l if a wrong path is followed. In this case (a wrong path is followed starting from the original node), $\sum_{l=0}^{l_n} d^2[z_l, a_l]$ can be lower bounded by the column distance function $d_{l_n}^2$ following (1.21) and $M(l_n)$ is upper bounded by

$$M(l_n) \leq -\alpha d_{l_n}^2 + \sum_{l=0}^{l_n} \beta(z_l). \tag{4.6}$$

A sequential decoder abandons a path whenever the Fano metric falls below the metric of a temporarily more likely path. From (4.5) it follows that a partial path has a small path metric and is rejected by the decoder if its distance from the received sequence is sufficiently large. But it is the speed of this rejection that determines the computational effort. Without loss of generality, we assume that the decoder follows a wrong path from the original node. Then, we have the upper bound of the path metric given by (4.6). The bound shows that the metric function along any path from the correct path decreases at least as fast as the column distance function grows. Thus, fast rejection of an incorrect path requires a rapidly decreasing metric along incorrect paths. Consequently, the rapidly increasing column distance function will guarantees fast decoding. This observation has long been recognized for convolutional codes [6, 8, 53]. From the above analysis, we see that similar conclusion can be drawn for trellis codes.

Let us present an example to verify this observation for trellis codes. Figure 4.2 shows the column distance functions (CDF's) of two trellis coded 8-PSK with a

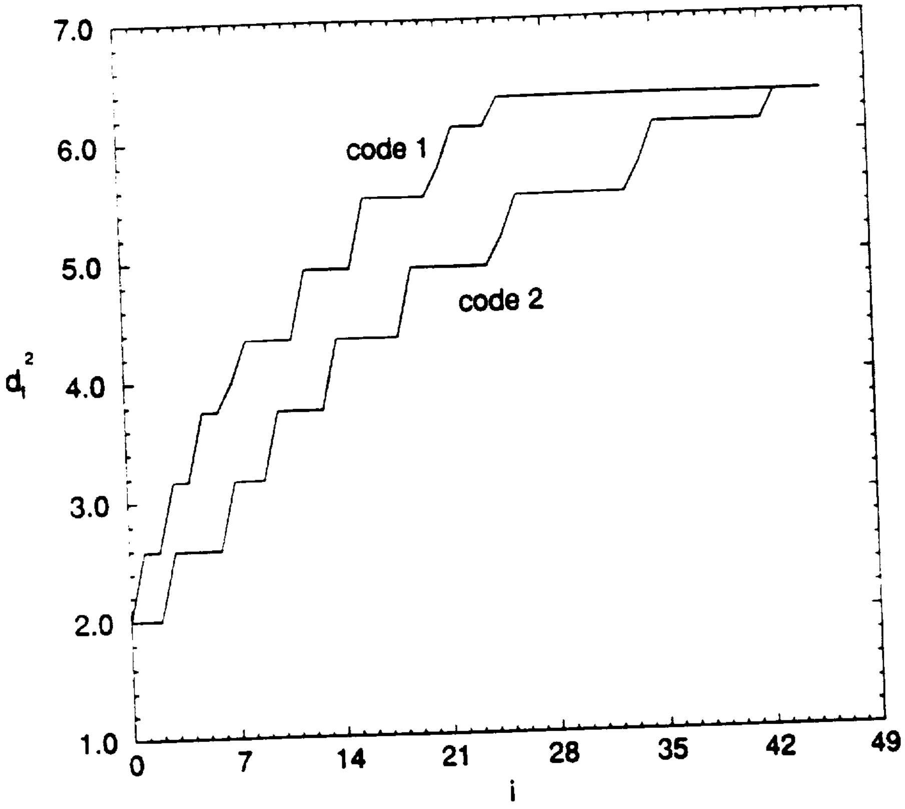


Figure 4.2: CDF's of two $\nu = 9$ trellis codes

constraint length $\nu = 9$. Code 1 has parity-check coefficients $H^0 = 1761$, $H^1 = 0106$, and $H^2 = 0400$ in octal form. The parity-check coefficients for code 2 are $H^0 = 1001$, $H^1 = 0036$, and $H^2 = 0546$. Both codes have the same free distance $d_{free}^2 = 6.343$. However, it is noted that the CDF of code 1 grows much faster than code 2 as shown in the figure. According to the above analysis, we may expect that the computational behavior of code 1 will be better than code 2. Figure 4.3 shows the computational distribution of the two codes at $SNR = 7.5$ dB. It is evident that the results are just as we expected, i.e., the computational behavior of code 1 is superior to code 2. This observation also holds for other SNR's. Figure 4.4 and Figure 4.5 show the computational distributions of the same two codes at $SNR = 8.0$ and 8.5 dB,

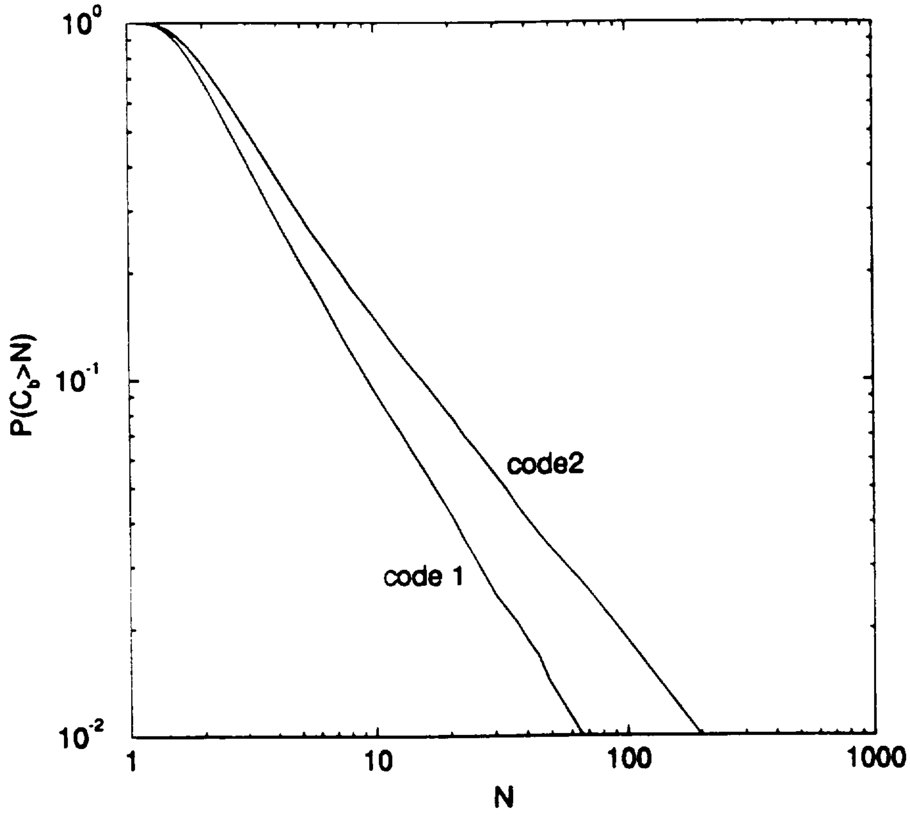


Figure 4.3: Computational distribution for sequential decoding of code 1 and code 2 at SNR=7.5 dB

respectively. It is seen that code 1 is much better than code 2 computationally.

The above analysis and simulation show that a trellis code with a rapidly growing CDF results in better computational effort for sequential decoding. Thus, trellis codes for use with sequential decoding should be designed such that their CDF's be optimized to minimize the computational effort. However, this approach is quite unrealistic since the number of distinct elements in a CDF is a random variable and is so large that it is impossible to evaluate every one of the CDF for large constraint length codes. Actually, it might not be necessary to optimize the CDF in the code construction for trellis codes by noting the following facts. First, from (4.6), we notice

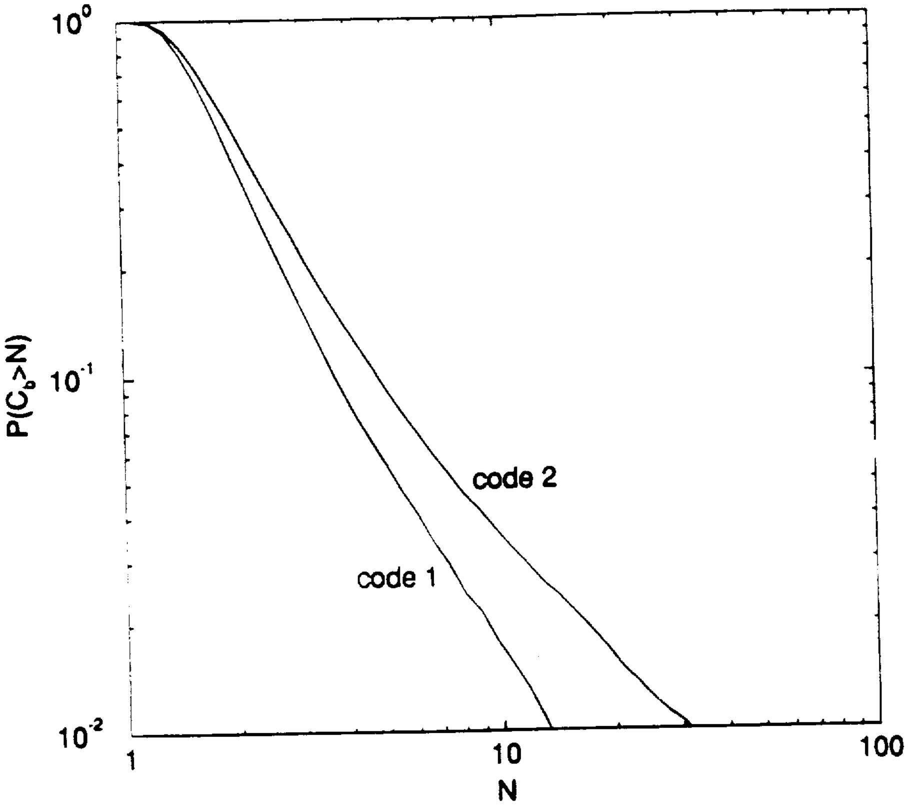


Figure 4.4: Computational distribution for sequential decoding of code 1 and code 2 at SNR=8.0 dB

:hat the initial portion of the CDF, i.e., the distance profile play a more important role than its later part since a faster growing initial portion will prevent the sequential decoder to get into a wrong path too deep and thus the computational effort to back down from the wrong path will be saved. Secondly, a code with a faster growing distance profile may also have a better CDF for a certain free distance. Thus, we may only need to optimize the distance profile in the code construction for sequential decoding. This approach has been used for construction of convolutional codes for sequential decoding[39]. To illustrate this point, we give one more example in this section.

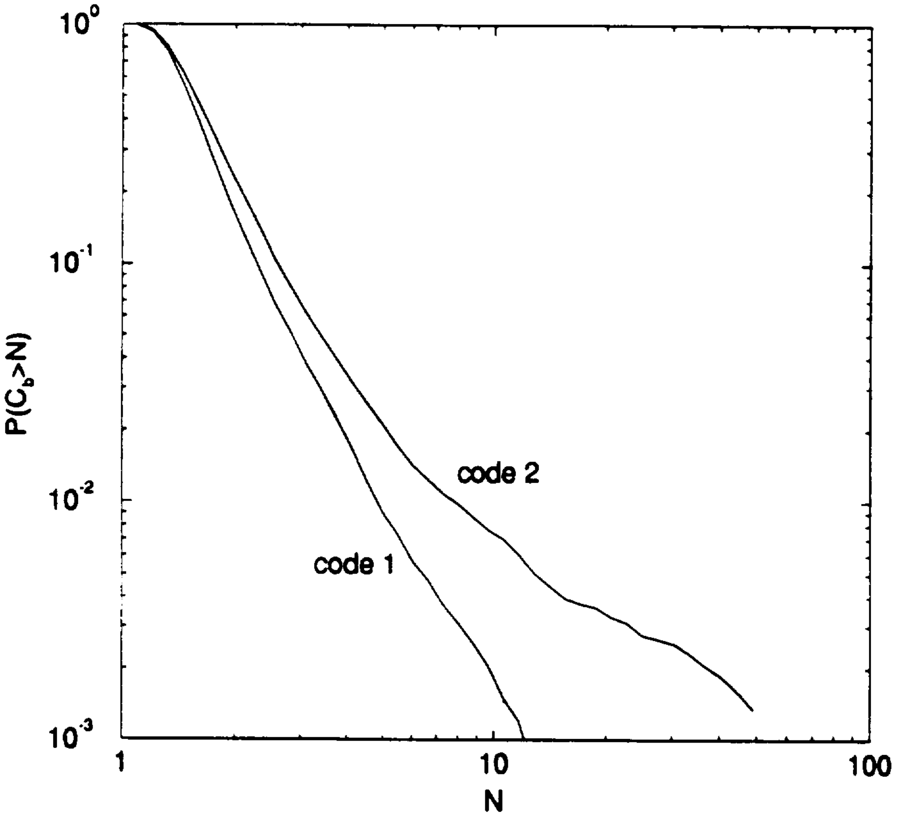


Figure 4.5: Computational distribution for sequential decoding of code 1 and code 2 at SNR=8.5 dB

Figure 4.6 shows the distance profiles of two trellis coded 8-PSK with a constraint length ν = 13. Code 3 is an ODP trellis code whose parity-check coefficients are H^0 = 33001, H^1 = 16266, and H^2 = 01400. Code 4 is the code constructed by Porath and Aulin[61] whose parity-check coefficients are H^0 = 20201, H^1 = 12746, and H^2 = 00200. Both codes have the same free distance d^2_{free} = 8.686. Figure 4.6 shows that code 3 has a more rapidly growing distance profile than code 4. Thus, the computational distribution of code 3 should be better than code 4. However, the difference of the distributions of the two codes may not as noticeable as code 1 and code 2 since the first several CDF's are identical. The computational distributions of

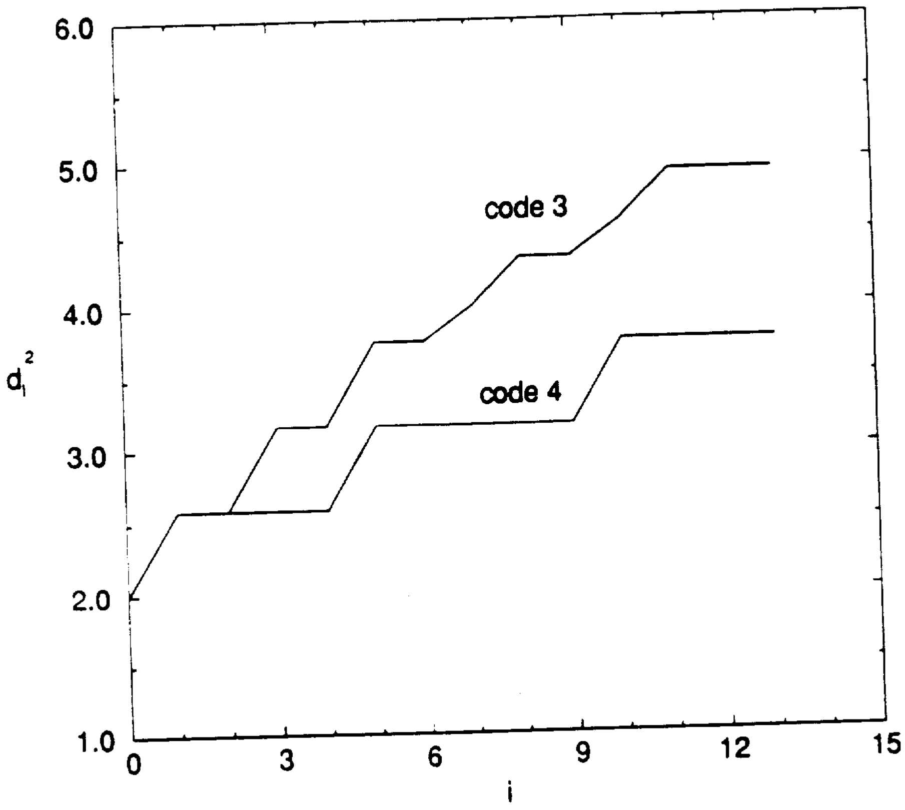


Figure 4.6: Distance profiles of two $\nu = 13$ trellis codes

the two trellis codes shown in Figures 4.7, 4.8, and 4.9 are just as we expected.

4.2 Optimum Distance Profile and Optimum Free Distance Trellis Codes

The above analysis and simulation results show that the column distance function, especially its initial portion, the distance profile, plays a very important role for sequential decoding. Trellis codes with good column distance functions or more importantly good distance profiles must be used for sequential decoding to achieve good

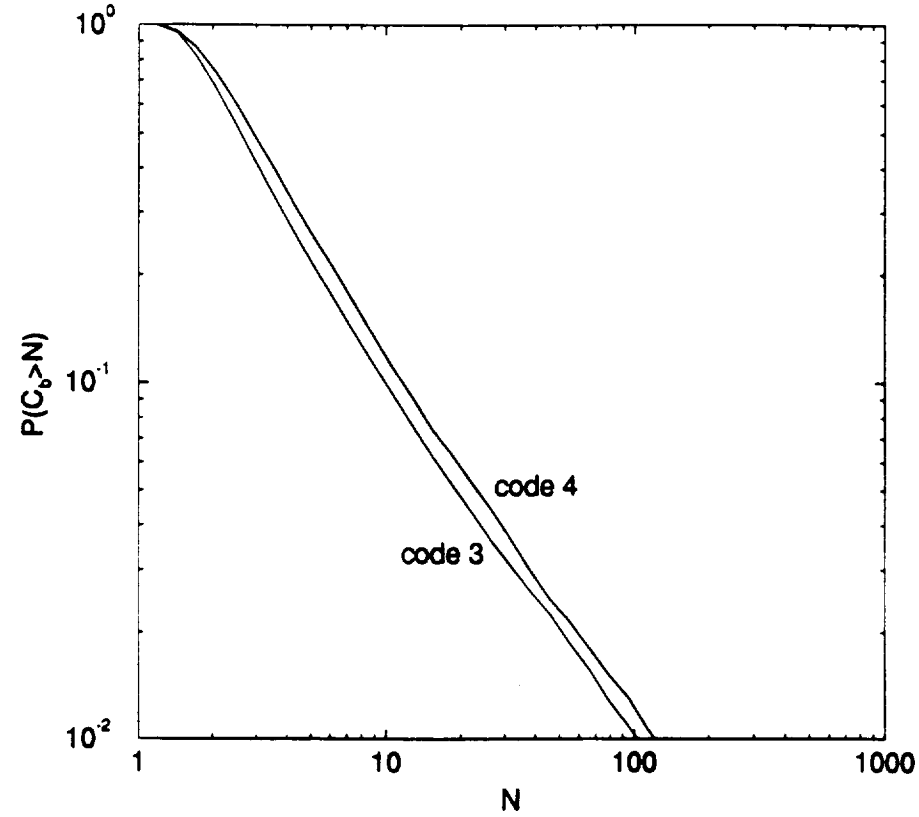


Figure 4.7: Computational distribution for sequential decoding of code 3 and code 4 at SNR=7.5 dB

computational performance. On the other hand, sequential decoding is a nearly maximum likelihood algorithm for which the error probability decreases exponentially with the free distance[8, 74]. Thus, we wish to maximize the free distance to reduce the error probability and to optimize the distance profile to achieve good computational performance.

A trellis code is said to have a distance profile $(d_0^2, d_1^2, \cdots, d_\nu^2)$ superior to the distance profile $(d_0'^2, d_1'^2, \cdots, d_\nu'^2)$ of another code of the same constraint length ν if for some p, $0 \leq p \leq \nu$,

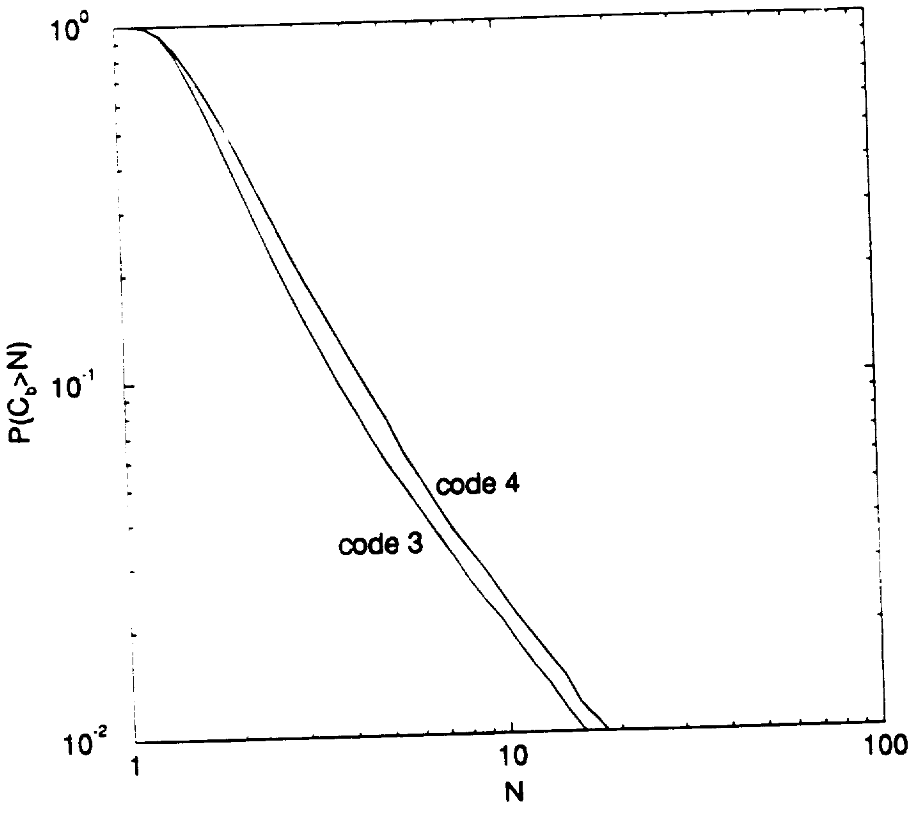


Figure 4.8: Computational distribution for sequential decoding of code 3 and code 4 at SNR=8.0 dB

$$\begin{aligned} d_i^2 &= d_i'^2, \quad i = 0, 1, \cdots, p-1 \qquad (4.7) \\ &> d_i'^2, \quad i = p. \end{aligned}$$

We say a code is an optimum distance profile code if its distance profile is equal or superior to that of any other code with the same constraint length. In this section, we present the computer search results for Optimum Distance Profile (ODP) and Optimum Free Distance (OFD) trellis codes for 8-PSK and 16-QAM modulation.

The code search algorithm is straightforward in an exhaustive search form. It retains the code which has the best distance profile. If several codes have the same

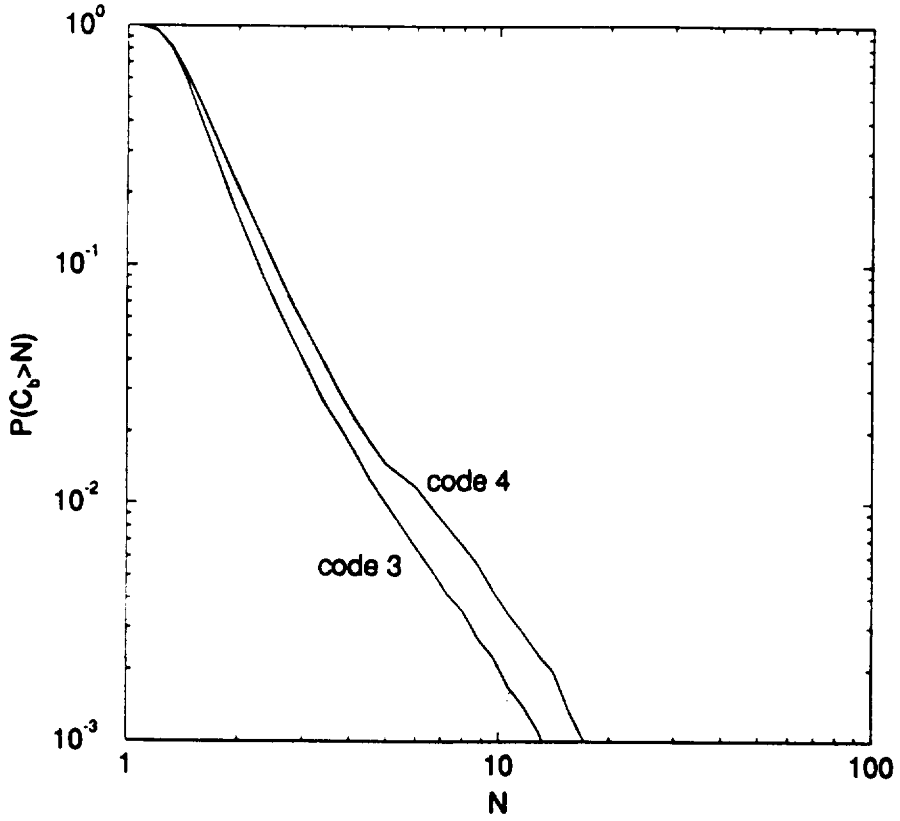


Figure 4.9: Computational distribution for sequential decoding of code 3 and code 4 at SNR=8.5 dB

distance profile, the one having the largest free distance is retained. The codes obtained using this approach may be called robustly optimal distance profile trellis codes following the notion used for convolutional codes[39]. We have observed that a lot of codes have the same distance profile but have a variety of free distances. For example, the free distances of ν = 13 trellis codes for 8-PSK modulation with an optimum distance profile are from 5.757 to 8.686. Thus, the construction of robustly optimal distance profile codes is necessary to guarantee finding codes that have large free distances.

Tables 4.1 and 4.2 show the results of computer searches for the ODP trellis codes

Table 4.1: ODP trellis codes for 8-PSK modulation

v	H^0	H^1	H^2	d_v^2				d_{free}^2				γ (dB)
				ODP	UG	P&A	MAL	ODP	UG	P&A	MAL	
3	15	02	-	3.17	2.59	-	3.17	4.00	4.59	-	3.17	3.01
4	25	16	-	3.17	2.59	2.59	3.17	4.00	5.17	5.17	3.17	3.01
5	77	26	-	3.76	3.17	3.17	3.76	4.00	5.76	5.76	3.76	3.01
6	177	042	-	3.76	2.59	2.59	3.76	4.00	6.34	6.34	3.76	3.01
7	201	072	-	4.00	2.59	2.59	4.00	4.00	6.59	6.59	4.00	3.01
8	701	266	300	4.34	2.59	2.59	4.34	6.93	7.52	7.52	4.34	5.40
9	1055	0502	0400	4.34	3.76	3.17	4.34	6.93	7.52	7.52	4.93	5.40
10	2201	0666	1400	4.59	3.17	3.17	4.34	7.52	7.52	8.10	4.93	5.75
11	4047	2302	0400	4.93	-	3.17	4.93	8.10	-	8.34	4.93	6.07
12	10517	06462	04400	4.93	-	3.76	4.93	8.34	-	8.69	4.93	6.20
13	33001	16226	01400	4.93	-	3.76	4.93	8.69	-	8.69	5.52	6.38
14	57001	22266	35400	5.52	-	-	4.93	8.69	-	-	5.52	6.38
15	104001	045666	035400	5.52	-	4.34	4.93	9.27	-	9.51	6.10	6.66

Table 4.2: ODP trellis codes for 16-QAM modulation

v	H^0	H^1	H^2	H^3	d_v^2			d_{free}^2			γ (dB)
					ODP	UG	MAL	ODP	UG	MAL	
3	11	06	–	–	4.0	3.0	3.0	4.0	5.0	3.0	3.01
4	23	12	–	–	4.0	3.0	4.0	4.0	6.0	5.0	3.01
5	61	12	20	–	4.0	3.0	4.0	5.0	6.0	5.0	3.98
6	115	016	020	–	4.0	4.0	4.0	6.0	7.0	5.0	4.77
7	261	132	100	–	5.0	3.0	5.0	7.0	8.0	5.0	5.44
8	401	066	100	–	5.0	4.0	5.0	8.0	8.0	6.0	6.02
9	1401	0166	0300	–	6.0	4.0	5.0	8.0	8.0	7.0	6.02
10	3101	1652	1500	–	6.0	–	5.0	8.0	–	7.0	6.02
11	4001	1352	1500	–	6.0	–	6.0	8.0	–	7.0	6.02
12	11657	06306	01300	–	6.0	–	6.0	8.0	–	7.0	6.02
13	31051	16606	15300	06000	6.0	–	6.0	9.0	–	8.0	6.53

for 8-PSK and 16-QAM modulations, respectively, where the parity-check coefficients H^j is defined as

$$H^j = (h^j_\nu, h^j_{\nu-1}, \cdots, h^j_0), j = 0, 1, \cdots, k. \tag{4.8}$$

All the H^j's are expressed in octal form. In the coding literatures, the as mptotic coding gain is often used to judge the "goodness" of a trellis code. Suppose that Δ_1 is the minimum distance between the points in a corresponding uncoded 2^k point constellation. The asymptotic coding gain γ of a trellis code compared to the uncoded case is given by

$$\gamma = 10 \log_{10}(d^2_{free}/\Delta^2_1) dB. \tag{4.9}$$

The free distance and the asymptotic coding gain γ of the optimum distance profile (ODP) codes are listed in the Tables.

It requires a large space to list the distance profile of a code. Thus, the minimum distances of the ODP codes, which are good indicators of the distance profiles, are listed in the Tables in stead of their distance profiles. For comparison, we also have included the minimum distance and free dist e of Ungerboeck (UG) codes [70, 72], Porath and Aulin's (P&A) codes [61], and the systematic feedforward codes constructed by Malladi et. al. (MAL) [49] in our Tables. Ungerboeck code and Porath and Aulin's codes are best known free distance trellis codes for two-dimensional modulations and the trellis codes of Malladi et. al. are the only trellis codes intended for use with sequential decoding which have good distance profiles. Comparison shows that the ODP trellis codes have much better distance profiles than the UG and P&A codes and slightly better than the MAL codes.

An anomaly shown in the Table 4.1 and 4.2 is that the free distances of some short constraint length ODP codes are much smaller than the UG and P&A codes. For example, the $\nu = 7$ ODP code in Table 4.1 has only a free distance of 4.0 compared with the free distance of 6.59 of the UG and P&A codes. Simple calculation shows that the $\nu = 7$ ODP code in Table 4.1 will lose 2.2 dB asymptotic coding gain compared with the UG and P&A codes that have a optimum free distance. This is quite different from the case of convolutional codes where the free distance suffers little when the distance profile is optimized[39]. Thus, ODP codes clearly do not provide the best trade-off between distance profile and free distance.

Next, we construct the trellis codes that have Optimum Free Distances (OFD). The code search algorithm is also straightforward in an exhaustive search form. The code search algorithm retains the code which has the best free distance. If several codes have the same free distances, the one having the best distance profile is retained. The codes obtained using this approach may be called robustly optimal free distance trellis codes also following the notion used for convolutional codes[39]. Tables 4.3 and 4.4 show the computer search results for the OFD trellis codes for 8-PSK and 16-QAM modulations, respectively. The notation is the same as in Table 4.1 and 4.2. For comparison, the minimum distance and free distance of Ungerboeck (UG) codes [70, 72], Porath and Aulin's (P&A) codes [61], and the systematic feedforward codes constructed by Malladi et. al. (MAL) [49] are also included in our Tables.

Tables 4.3 and 4.4 show that the OFD codes achieve a much better distance profile than The *UG* and *P&A* codes. However, compared them with the ODP trellis codes, it seems that the OFD trellis codes do not provide the best trade-off between distance profile and free distance either. We give an example to illustrate this point. Figure 4.10 shows the distance profiles of ODP, OFD, and Ungerboeck (UG) trellis coded 8-PSK with $\nu = 7$. It shows that the OFD code has a much inferior distance profile to the ODP code although it improves upon the Ungerboeck code.

From the above discussion, we may conclude that neither the ODP nor the OFD

Table 4.3: OFD trellis codes for 8-PSK modulation

v	H^0	H^1	H^2	d_v^2				d_{free}^2				γ (dB)
				OFD	UG	P&A	MAL	OFD	UG	P&A	MAL	
3	13	02	04	2.59	2.59	-	3.17	4.59	4.59	-	3.17	3.61
4	35	02	10	3.17	2.59	2.59	3.17	5.17	5.17	5.17	3.17	4.12
5	67	02	34	3.17	3.17	3.17	3.76	5.76	5.76	5.76	3.76	4.59
6	147	042	060	3.17	2.59	2.59	3.76	6.34	6.34	6.34	3.76	5.01
7	375	064	112	3.17	2.59	2.59	4.00	6.59	6.59	6.59	4.00	5.18
8	515	356	314	3.17	2.59	2.59	4.34	7.52	7.52	7.52	4.34	5.75
9	1201	0666	0300	4.34	3.76	3.17	4.34	7.52	7.52	7.52	4.93	5.75
10	2771	1112	0400	4.34	3.17	3.17	4.34	8.10	7.52	8.10	4.93	6.07

trellis codes provide good compromise between distance profile and free distance for some constraint length. In next section, we present an algorithm to construct robustly good trellis codes which provide a good trade-off between distance profile and free distance.

4.3 Robustly Good Trellis Codes

It has been shown that both the ODP and the OFD trellis codes might not be the good choice for use with sequential decoding for some constraint lengths. In this section, we propose a new construction algorithm to construct trellis codes that have both a good distance profile and a good free distance. Actually, some of the codes

Table 4.4: OFD trellis codes for 16-QAM modulation

v	H^0	H^1	H^2	d_v^2			d_{free}^2			γ (dB)
				OFD	UG	MAL	OFD	UG	MAL	
3	15	02	04	3.0	3.0	3.0	5.0	5.0	3.0	3.98
4	31	12	14	3.0	3.0	4.0	6.0	6.0	5.0	4.77
5	41	22	34	4.0	3.0	4.0	6.0	6.0	5.0	4.77
6	121	012	054	4.0	4.0	4.0	7.0	7.0	5.0	5.44
7	325	056	120	4.0	3.0	5.0	8.0	8.0	5.0	6.02
8	501	166	100	5.0	4.0	5.0	8.0	8.0	6.0	6.02
9	1401	0266	0300	6.0	4.0	5.0	8.0	8.0	7.0	6.02

obtained using this algorithm have an optimum distance profile or an optimum free distance. However, the algorithm itself does not guarantee finding an optimum code in any sense. We call the codes constructed using this algorithm Robustly Good Codes (RGC).

Assume that a robustly good trellis code of constraint length ν is obtained. The approach used to find a constraint length $\nu+1$ robustly good trellis code is to find the code that improves the free distance or the distance profile of the constraint length ν code with the priority of improving the free distance. In other words, we try to find a longer code which has a free distance or a distance profile superior or identical to the shorter one.

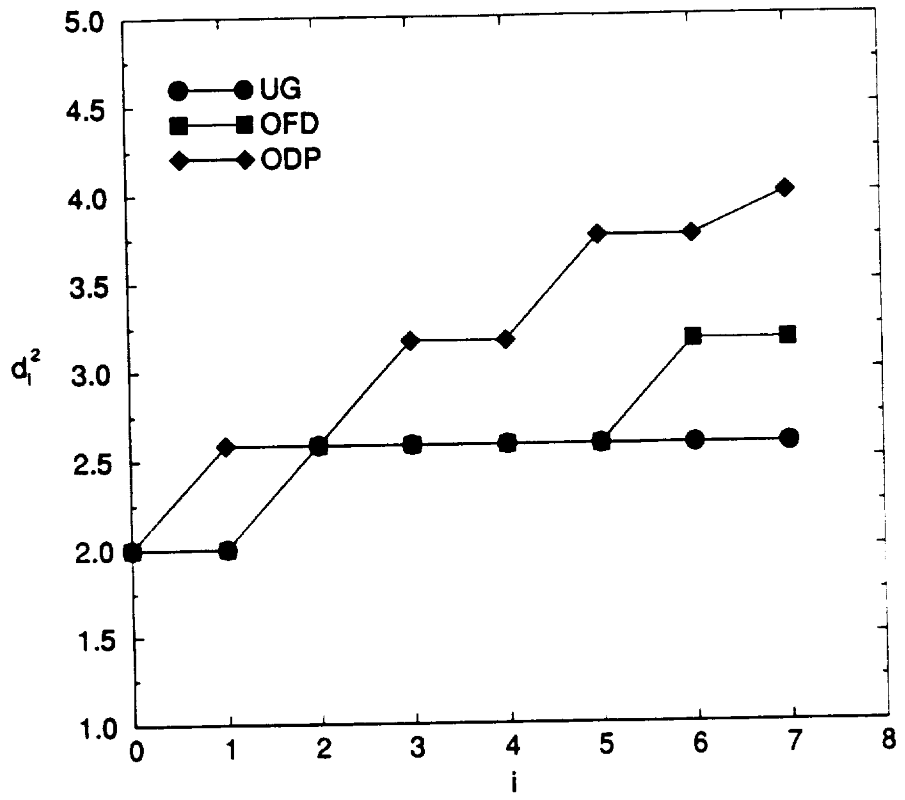


Figure 4.10: Comparison of distance profiles for three $\nu = 7$ trellis codes

Suppose that the free distance and distance profile of a robustly good trellis code with constraint length ν are $d_{free}^2(\nu)$ and $d^2(\nu) = \{d_0^2(\nu), d_1^2(\nu), \cdots, d_\nu^2(\nu)\}$, respectively. Then a robustly good trellis code with constraint length $\nu + 1$ can be found using the following algorithm:

0) Set $d_{free}^{2\prime} = d_{free}^2(\nu)$ and $d^{2\prime} = \{d_0^{2\prime}, d_0^{2\prime}, \cdots, d_\nu^{2\prime}, d_{\nu+1}^{2\prime}\} = \{d_0^2(\nu), d_1^2(\nu), \cdots, d_\nu^2(\nu), d_\nu^2(\nu)\}$.

1) Select a new code C by systematically changing the parity-check coefficients. Set $i = 0$.

2) Compute the column distance d_i^2 of code C.

3) If $d_i^2 < d_i^{2\prime}$, go to 8). Otherwise $i \leftarrow i + 1$, go to 4).

4) If $i \leq \nu + 1$, go to 2). Otherwise, go to 5).

5) Compute the free distance d^2_{free} of code C. If $d^2_{free} > d^{2\prime}_{free}$, print the parity-check coefficients of code C, d^2_{free}, $d^2 = \{d^2_0, d^2_1, \cdots, d^2_{\nu+1}\}$, and "a better free distance code is found". Otherwise, go to 6).

6) If $d^2_{free} < d^{2\prime}_{free}$, go to 8). Otherwise, go to 7).

7) If $d^2_i > d^{2\prime}_i$ for some i, print the parity-check coefficients of code C, d^2_{free}, d^2, and "a better distance profile code is found".

8) If the set of codes is exhausted, stop. Otherwise, go to 1).

A code will be excluded for further consideration if it has a inferior distance profile to the tentative distance profile $d^{2\prime}$. Since $d^{2\prime}$ is very close to the optimum distance profile, majority of codes will be eliminated before the algorithm reaches step 4). The tentative distance profile is updated only if a new code with the same or better distance profile has a larger free distance than the free distance. This is different from the search for ODP codes where the distance profile is updated whenever a new tentative code has a better distance profile, which guarantees finding codes with optimum distance profile but may result in codes with poor free distance as shown in Table 4.1 and 4.2. The algorithm also prints out those codes that have the same free distance as the tentative code $d^{2\prime}_{free}$ but a better distance profile. This allows us to select the codes that result in a good trade-off between the distance profile and free distance. Thus the algorithm guarantees finding a trellis code that is no worse than its previous constraint length in terms of free distance and distance profile.

The initial code can be chosen such that it results in a good trade-off between the distance profile and free distance. We start to construct trellis codes at a constraint length of 3. We retained a code that has the same free distance and the same distance profile as the OFD code. The trellis codes with larger constraint lengths are then constructed using the above algorithm. Tables 4.5 and 4.6 show the results of computer searches for the robustly good trellis codes for 8-PSK and 16-QAM modulations, re-

Table 4.5: Robustly good trellis codes for 8-PSK modulation

ν	H^0	H^1	H^2	d_v^2				d_{free}^2				γ (dB)
				RGC	UG	P&A	MAL	RGC	UG	P&A	MAL	
3	15	06	04	2.59	2.59	–	3.17	4.59	4.59	–	3.17	3.61
4	35	12	10	3.17	2.59	2.59	3.17	5.17	5.17	5.17	3.17	4.12
5	67	26	20	3.17	3.17	3.17	3.76	5.17	5.76	5.76	3.76	4.12
6	121	066	060	3.76	2.59	2.59	3.76	6.00	6.34	6.34	3.76	4.77
7	337	026	100	3.76	2.59	2.59	4.00	6.34	6.59	6.59	4.00	5.01
8	701	166	300	4.34	2.59	2.59	4.34	6.93	7.52	7.52	4.34	5.40
9	1175	0142	0400	4.34	3.76	3.17	4.34	6.93	7.52	7.52	4.93	5.40
10	2015	0402	0400	4.34	3.17	3.17	4.34	7.76	7.52	8.10	4.93	5.89
11	4047	2302	0400	4.93	–	3.17	4.93	8.10	–	8.34	4.93	6.07
12	10517	06462	04400	4.93	–	3.76	4.93	8.34	–	8.69	4.93	6.20
13	33001	16226	01400	4.93	–	3.76	4.93	8.69	–	8.69	5.52	6.38
14	57001	22266	35400	5.52	–	–	4.93	8.69	–	–	5.52	6.38
15	104001	045666	035400	5.52	–	4.34	4.93	9.27	–	9.51	6.10	6.66

Table 4.6: Robustly good trellis codes for 16-QAM modulation

v	H^0	H^1	H^2	H^3	d_v^2			d_{free}^2			γ (dB)
					RGC	UG	MAL	RGC	UG	MAL	
3	13	02	04	–	3.0	3.0	3.0	5.0	5.0	3.0	3.98
4	23	06	14	–	3.0	3.0	4.0	6.0	6.0	5.0	4.77
5	41	16	34	–	4.0	3.0	4.0	6.0	6.0	5.0	4.77
6	141	046	054	–	4.0	4.0	4.0	7.0	7.0	5.0	5.44
7	315	102	120	–	4.0	3.0	5.0	8.0	8.0	5.0	6.02
8	441	372	300	–	5.0	4.0	5.0	8.0	8.0	6.0	6.02
9	1401	0166	0300	–	6.0	4.0	5.0	8.0	8.0	7.0	6.02
10	2501	1352	1500	–	6.0	–	5.0	8.0	–	7.0	6.02
11	4001	3352	1500	–	6.0	–	6.0	8.0	–	7.0	6.02
12	16547	05326	04300	–	6.0	–	6.0	8.0	–	7.0	6.02
13	35153	06452	13500	16000	6.0	–	6.0	9.0	–	8.0	6.53

spectively. For comparison, we also have included the minimum distance and free distance of Ungerboeck (UG) codes [70, 72], Porath and Aulin's (P&A) codes [61], and the systematic feedforward codes constructed by Malladi et. al. (MAL) [49] in our Tables. Compared with Table 4.1 - 4.4, we see that the new codes achieve nearly the same free distances as the OFD codes and nearly the same distance profiles as the ODP codes.

4.4 Simulation Results

Robustly good trellis codes have been constructed in last section. Comparison of the new codes with the best known trellis codes of Ungerboeck[70] and Porath and Aulin[61] shows that the new codes provide a better trade-off between distance profile and free distance. In this section, simulation results are presented to show that better performance can be achieved using sequential decoding when these codes are used.

First, we note that the error performance of a trellis code is primarily determined by its distance spectrum when a maximum likehood (such as Viterbi) decoding is used[74, 95]. At high SNR's, the dominant term in the distance spectrum is the free distance and its multiplicities. Thus, the trellis codes that have the largest free distances and the smallest multiplicities are usually chosen for use with Viterbi decoding. It has been shown[76, 94] that sequential decoding can perform about as well as Viterbi decoding by random coding arguments. It has been further shown [8] that the error performance of sequential decoding for a specific code is also determined by its free distance and its multilicities. Thus, we expect that the (undetected) error performance of sequential decoding for a trellis code be determined by the code's free distance and its multiplicities.

The undetected error probabilities of the new codes and the Porath and Aulin's codes[61] are compared first. To obtain the undetected bit error rate, we allow a Fano sequential decoder to decode a noisy sequence without a time limit (or with an infinite buffer size). In Figure 4.11, we show the error performance for sequential decoding

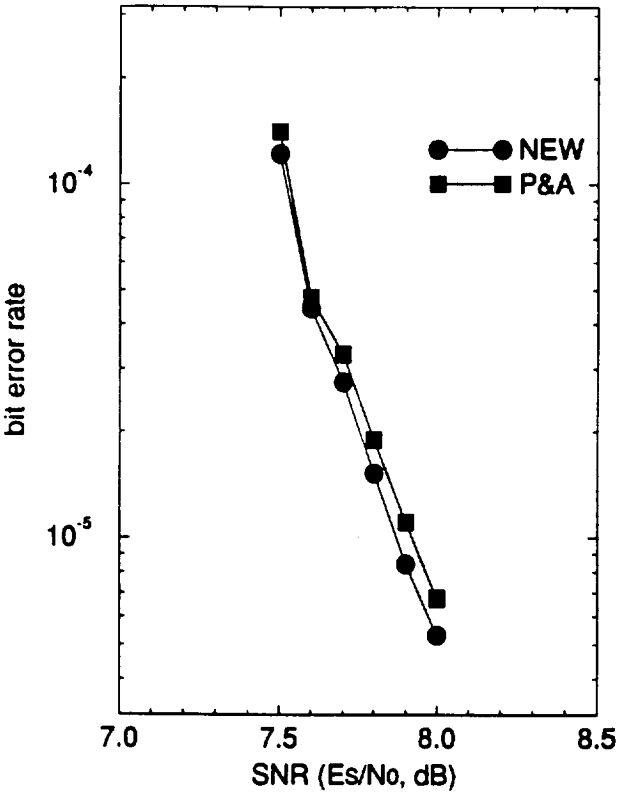


Figure 4.11: Performance comparison using sequential decoding

of the new (NEW) rate 2/3 trellis coded 8-PSK taken from Table V and the Porath and Aulin's (P&A) rate 2/3 trellis coded 8-PSK with constraint length $\nu = 13$. Both the NEW code and the P&A code have the same free distance $d^2_{free} = 8.69$. Thus, we expect that both NEW and P&A codes have about the same error performance. Figure 4.11 shows that the NEW code actually perform better than the P&A code. Intensive simulations have been run for other constraint lengths. Results show that the codes that have a substantially better column distance function or distance profile always outperform other codes in term of undetected error probability when the free distances are comparable. This may be attributed to that the early growth of the column distance function can prevent the sequential decoder to follow a wrong path too deep to be get rid of it.

The above simulation results and discussion show that the new codes achieve better (undetected) error performance than the best known codes for the same constraint length when sequential decoding is used. We also have demonstrated that the new codes can achieve better computational performance since they have better distance profiles. In practical application of sequential decoding, the data can be transmitted in blocks. Some very noisy blocks may require a large amount of computations which may be untolerable in practice. Sequential decoder may be applied in an ARQ communication system [12]. In such a system, a block can be declared unreliable and a retransmission can be requested if a predetermined computational limit is exceeded for a block. In an ARQ system, the system throughput, which is defined as the number of blocks that are successfully received over the total number of blocks attempted, is the primary performance criterion. The better computational performance of the new codes with sequential decoding implies larger throughput. On the other hand, Figure 4.11 shows that the new codes achieve better error probability than other codes that known to us. Thus, the new codes are compared very favorably with the best known codes in an ARQ communication system.

In Chapter 3, we showed that the Buffer Looking Algorithm (BLA) can be used to achieve erasurefree sequential decoding. In this case, the BLA decoder buffer will less likely be occupied if the new codes are used because of their superior computational performance. Thus, better overall performance can be achieved when the new codes are used with the BLA. In Figure 4.12, we compare the overall performance of the same NEW code and P&A code as in Figure 4.11 with constraint length of $\nu = 13$ using the Buffer Looking Algorithm. The BLA-BD with a buffer size $B = 16$ K

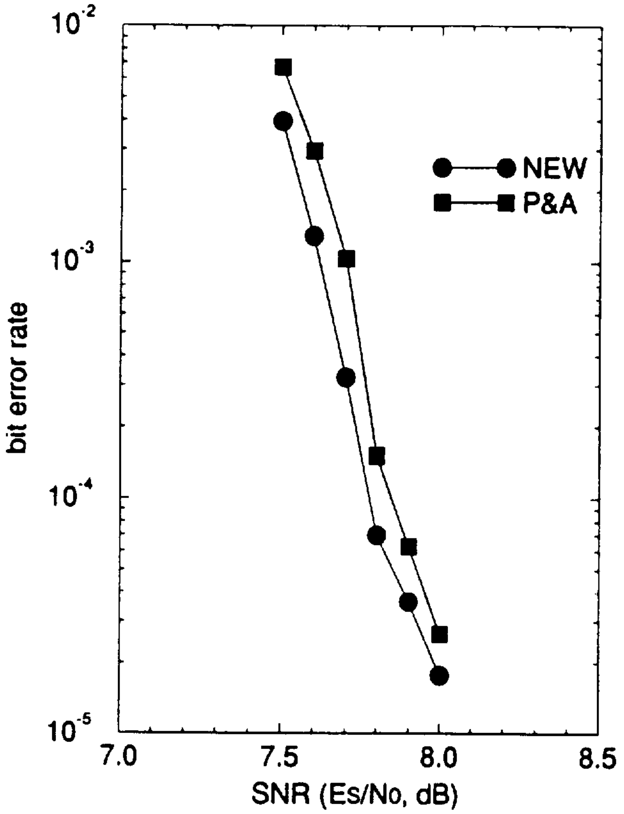


Figure 4.12: Performance comparison using the BLA

symbols, a decoder speed factor $\mu = 4$, and a block length $L = 256$ symbols (512 information bits) was used for our simulation. Figure 4.12 shows that the NEW code has a better overall performance over the P&A and the new code achieves about 0.1 dB coding gain over the Porath and Aulin code for the same constraint length. The number may not be very impressive, but we must keep in mind that this gain comes out without any other penalty. It is the pure gain by using our new codes for sequential decoding.

5

PROBABILISTIC CONSTRUCTION OF TRELLIS CODES

In Chapter 4, optimum or nearly optimum trellis codes are constructed for use with sequential decoding. Trellis codes for 8-PSK and 16-QAM modulation with constraint lengths up to 15 have been found. These codes have been shown to perform better than the best known trellis codes when sequential decoding is used. However, the code construction algorithms used are essentially exhaustive search with some rejection rules. The number of possible codes for a rate $k/k+1$ systematic feedback code with constraint length ν is about $2^{(k+1)\nu}$. Thus, it becomes impractical to conduct exhaustive search for large constraint lengths.

Porath and Aulin[61] proposed non-exhaustive search code construction algorithms for construction of good large constraint length trellis codes. Their algorithms are a generalization and combination of Lin and Lyne [10, 33, 47] type algorithm. Malladi et al [49] also used a Lin and Lyne type algorithm to construct systematic feedforward trellis codes for use with sequential decoding. This type of algorithms guarantee that codes with good distance growth can be found and thus they appear to be a good choice for construction of convolutional or trellis codes for use with sequential decoding. However, it is the code free distance that determines the error performance. Lin and Lyne type algorithm cannot guarantee that codes with large free distances are found. Furthermore, it is very difficult to evaluate the free distance of codes with large constraints. This poses a problem for the selection of good codes in any conventional code construction algorithm. In this chapter, we investigate a probabilistic approach to construct good large constraint length trellis codes for use with sequential decoding. In Section 5.1, results from random coding are reviewed and simulation results for trellis codes are presented to illustrate how randomly chosen codes perform. In Section 5.2, two code construction algorithms are proposed. In section 5.3, simulation results are presented to show that the codes constructed can achieve the cut-off rate bound at a bit error rate of $10^{-5} - 10^{-6}$.

5.1 Results from Random Coding

Traditionally, trellis codes are selected based upon either the free Euclidean distance, distance spectrum, and/or the distance profile depending on whether Viterbi decoding or sequential decoding is used for decoding. Actually, the error performance of a code can only be determined by its entire distance spectrum. Better free distance may not result in better performance since the multiplicity of the free distance and of some larger distances also play an important role. For trellis codes, the difference between the free distance and the next smallest distance of a code may be very small [66, 77]. Hence, using free distance as the only measure for selecting good codes may not be justified even for Viterbi decoding.

The ultimate purpose of code construction is to determine the codes that give the best performance. For large constraint length codes, it may be easier to select codes based upon their actual performance than upon their distance spectrum. SD performs almost as well as the VA and its average number of computations is very small. Thus, it is an ideal tool to examine the performance of a set large constraint length codes during code construction. Since it is virtually impossible to calculate the free distance of large constraint length codes, much less than the entire distance spectrum, the evaluation of the performance of large constraint length codes may be the best practical way to construct large constraint length codes.

Small constraint length codes may be constructed either by hand or by exhaustive search. However, since the number of possible codes increases exponentially with the constraint length, it is impossible to conduct an exhaustive search to construct large constraint length codes. Large constraint length codes are usually constructed by restricting the search to a small set of codes[49, 61]. It is well known that the average error probability of all rate $R = k/n$ trellis codes satisfies the bound[76]

$$P_{av}(e) < (2^k - 1)\frac{2^{-(\nu+1)kR_0/R}}{[1 - 2^{-\epsilon kR_0/R}]^2} \tag{5.1}$$

for $0 \leq R \leq R_0(1 - \epsilon)$, where ν is the constraint length of the code, ϵ is a positive constant, and R_0 is the computational cut-off rate of sequential decoding. It can be shown that at least a fraction $1 - \lambda$ of all codes in the collection must have a $P(e)$ no larger than $\frac{1}{\lambda}P_{av}(e)$ [89]. For example, it is shown that at least 90% of all codes have error probability $P(e) < 0.1P_{av}(e)$ or 50% of all codes have $P(e) < 0.5P_{av}(e)$ by choosing $\lambda = 0.1$ or 0.5 respectively.

Although the random coding bound (5.1) is derived for convolutional coding, we feel strongly that it still holds for trellis coding since the two of them is very similar in the sense of code structure. To see what the random coding bound means, performance of some randomly chosen trellis codes are evaluated by sequential decoding. Figure 5.1 shows the performance of 200 randomly chosen rate 2/3 trellis codes for 8-PSK modulation with $\nu = 8$ at $SNR = 8.0$ dB. It is noted that several codes are found with very good performance and most of the codes perform very close. Similar results can be gotten for codes with QAM modulation. Figure 5.2 shows the perfor-

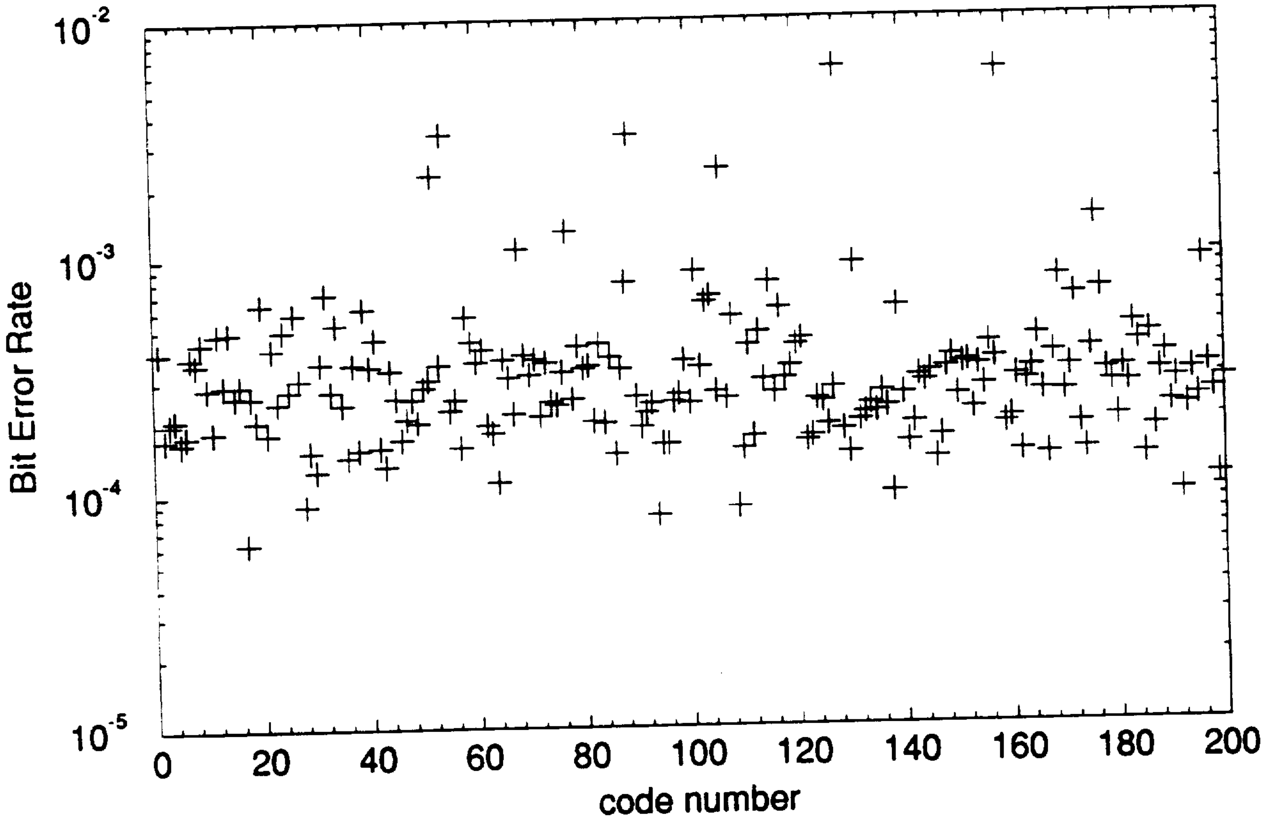


Figure 5.1: Performance of randomly chosen 8-PSK codes

mance of 200 randomly chosen rate 3/4 trellis codes for 16-QAM modulation with $\nu = 8$ at $SNR = 11.5$ dB.

The above discussions indicate that many good codes exist. Hence if the best codes cannot be found, a randomly selected code will probably give good error performance. From (5.1), it is seen that arbitrarily low error probability can be achieved with sufficiently large ν. Since the computational effort of sequential decoding is essentially independent of ν, sequential decoding can achieve very good performance with tolerable complexity when the code rate R is less than the cut-off rate R_0.

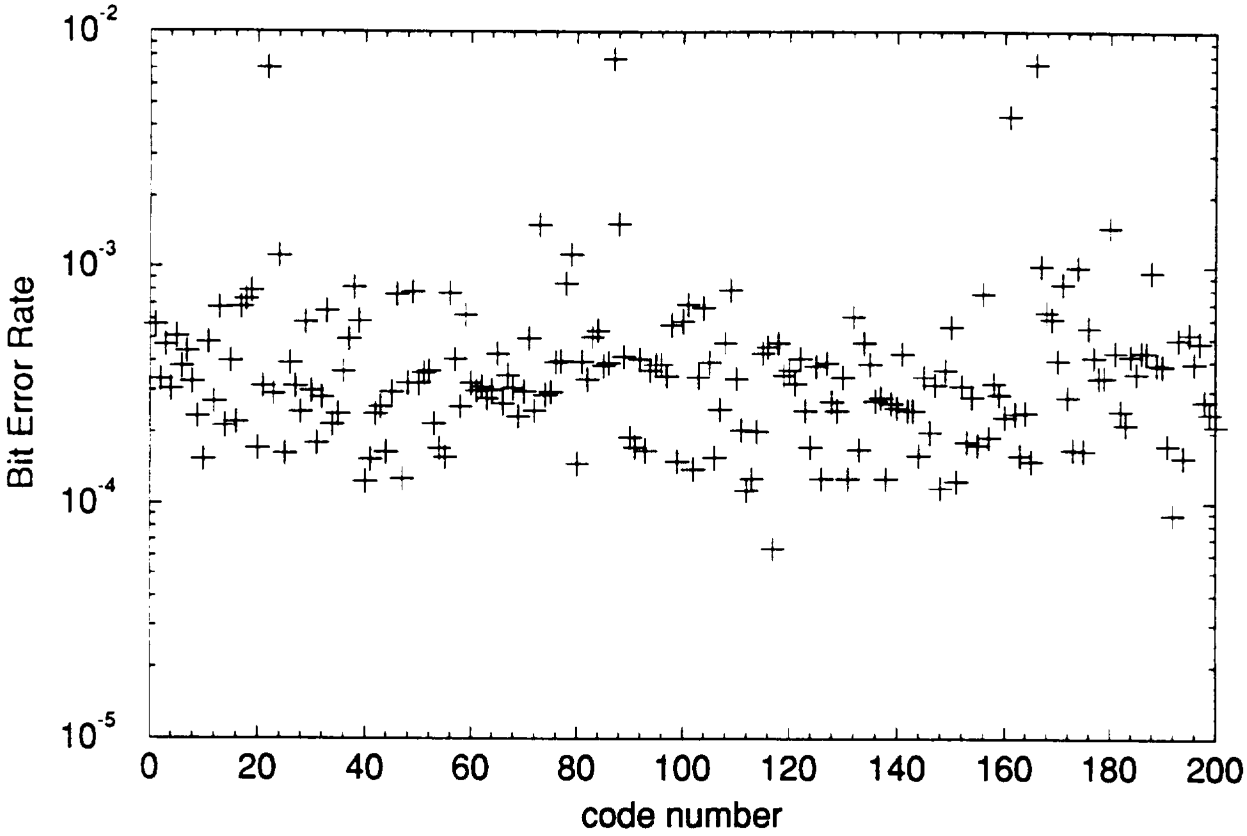


Figure 5.2: Performance of randomly chosen 16-QAM codes

5.2 Code Construction

Analysis and simulation results in last section show that many good codes exist. In this section, we present two construction algorithms to search good large constraint length trellis codes. The parity check coefficients of trellis codes are generated randomly and the performance of the codes are evaluated and compared. Good codes are retained. The two algorithms differ in the way to stop the code search process. The first algorithm stops after a certain number of codes are examined and the second algorithm utilizes simulated annealing approach to stop the process. Our construction algorithms restrict the search to a small set of codes just like those of [49, 61], but the set of codes are chosen randomly instead.

Let N_c be the number of codes to be examined, N_b be the number of encoded sequences (each sequence consists of m information bits) to be decoded for each code, and P_b be the average bit error probability of a code. The first construction algorithm is as follo'vs.

Construction Algorithm 1 (CA-1, Random Search):

1. Choose the SNR at which the codes are to be evaluated, N_c, and N_b. Let n_c and n_b be the number of codes examined and sequences decoded thus far, respectively. Set $n_c = 0$, $n_b = 0$, and $P_b = 1.0$.

2. Select a code by randomly choosing the generator (or parity-check) coefficients.

3. Encode a randomly chosen sequence of m information bits using the code chosen in 2.

4. Add channel noise to the encoded sequence.

5. Decode the corrupted sequence using sequential decoding. Set $n_b = n_b + 1$. If $n_b < N_b$, go to 3. Otherwise, go to 6.

6. Calculate the average bit error probability P_{bt} of the N_b encoded sequences. If $P_{bt} > P_b$, go to 8. If $P_{bt} \leq P_b$, go to 7.

7. Print P_{bt} and the generator (or parity-check) coefficients of the code. Set $P_b = P_{bt}$.

8. Set $n_c = n_c + 1$. If $n_c < N_c$, go to 2. Otherwise, stop.

Our confidence in the performance evaluation of a code depends on the number of errors decoded. It requires decoding more encoded sequences to make more errors. This results in longer computer search time. N_b is usually chosen to make sure that several hundred errors being decoded. Large SNR results in few errors. Thus, longer computer search time is required to generate a fixed number of errors for larger SNR. Usually, the codes constructed at a low SNR have small multiplicities. As we will show below, they perform better than the Ungerboeck codes at low SNR. In our code construction, a SNR close to cut-off rate bound is chosen. The information block size m is chosen to be 1000 and 1500 bits for 8-PSK and 16-QAM modulation respectively.

Some modifications to the above algorithm can be made to speed up the construction. The BER of a good constraint length ν code cannot be larger than the BER of a good $\nu - 1$ code. Thus, we get an estimate of the expected number of errors for a constraint length ν code from previous constructions. The estimate can be used as a limit for the number of errors. When the number of decoded errors for a code exceeds the limit, the performance evaluation can be stopped and this code is eliminated as a bad code. Similarly, we can also set a limit for the average number of computations. Once the average number of computations for a code exceeds the limit, the performance evaluation can be stopped and the code is eliminated as a bad one. These modifications drop some poor codes in an earlier stage of performance evaluation. Thus, computer search time can be reduced and the performance of the codes constructed will not be affected.

To insure that good codes are found, two steps are employed in our construction. First, several codes that perform well at the chosen SNR are obtained from CA-1. These codes are then evaluated over a wide range of SNR's with much more data being decoded. This allows us to select the best code with a high degree of confidence while the computer search time is reduced significantly.

Code construction may also be viewed as a combinatorial optimization problem where the parity check (or generator) coefficients are the variables and the free distance or the performance of a code is the objective (cost) function. A typical combinatorial optimization problem seeks the minimum or maximum of a given objective (or cost) function of many variables. The objective function represents a quantitative measure of the "goodness" (or "badness") of some complex system. The variables are subject to interwining constraints.

Simulated annealing is a computational heuristic for obtaining approximate solutions to combinatorial optimization problems. Initially, the Metropolis algorithm [55] was used to simulate numerically the annealing process to gain an understanding of the ground state configuration. Kirkpatric et al. [42] first investigated the use of simulated annealing in connection with the physical design of computers. Since then, it has been applied to various combinatorial problems with varying degree of success [13, 15]. Good block codes (both source and channel codes) have been constructed using simulated annealing in [13]. We investigate the construction of good trellis codes using simulated annealing.

Define the energy (cost function) of a code C as $E(C) = P_b(C)$, where $P_b(C)$ is the average bit error probability of the code C at some SNR. Let N_e be the number of energy drops required to lower the temperature, N_i be the number of iterations required to lower the temperature, and N_c be the number of consecutive temperature stages that produce no change in the code required to stop the code search. The construction algorithm is as follows.

Construction Algorithm 2 (CA-2, Simulated Annealing):

1. Let n_e be the number of energy drops, n_i be the number of iterations, and n_c be the number of consecutive temperature stages that produce no change in the code. Choose a code C and a temperature T. Let $n_e = 0$, $n_i = 0$, and $n_c = 0$.

2. Choose :: code C', a perturbation of C (randomly "jiggle" one coefficient). Let $\Delta E = E(C') - E(C)$. If $\Delta E < 0$, $C \leftarrow C'$ and $n_e = n_e + 1$. Otherwise, with probability $\exp(-\Delta E/T)$, $C \leftarrow C'$. If $C \leftarrow C'$ occurs, let $n_c = 0$.

3. $n_i = n_i + 1$.

4. If $n_e \geq N_e$, go to 6. Otherwise, go to 5.

5. If $n_i \geq N_i$, go to 6. Otherwise, go to 2.

6. Let $n_e = 0, n_i = 0, n_c = n_c + 1$, and T $\leftarrow \alpha T$ $(1 > \alpha \geq 0.9$, a constant). If $n_c < N_c$, go to 2. Otherwise, print out the code generator (or parity-check) coefficients and stop.

A code with all zero coefficients (a poor code) is chosen as the initial C. T is initially chosen to be roughly one hundred times the expected BER of the best code. In the beginning, almost all perturbations are accepted. However, as the temperature is reduced, the acceptance probability will be lowered. The temperature is reduced by a factor $\alpha = 0.9$ after three energy drops $(N_e = 3)$ or after more than 20 $(N_i = 20)$ perturbations, whichever comes first. The algorithm terminates if five $(N_c = 5)$ consecutive temperature stages do not produce any change in the code. The choice of parameters is obtained by experimentation. With the above parameters, the algorithm is usually terminated after several hundred to several thousand codes are searched and seems to yield satisfactory results.

To compare the two construction algorithms, trellis codes for 8-PSK modulation with constraint length $\nu = 7$ and 8 are constructed. A total of two hundred codes are evaluated for the CA-1 while about one thousand codes are evaluated for the CA-2. The codes are evaluated at a SNR=7.75 dB which is slightly larger than the R_0^* bound. The performance of the best codes is compared in Figure 5.3. It shows that the codes constructed by the two algorithms perform almost the same. Since it is much faster, the CA-1 is used in this paper for our code search.

Tables 5.1 and 5.2 show the computer search results for trellis codes with 8-PSK and 16-QAM modulation where the row parity check H^i is defined as in Chapter 4. All the H^i's are expressed in octal form. The performance of the codes are evaluated by sequential decoding. The real coding gain of the new (NEW), Ungerboeck (UG), and Porath and Aulin (P&A) codes at a BER of 10^{-5} over an uncoded system are also listed in the tables. The real coding gain is defined as the difference between

Table 5.1: Trellis codes for 8-PSK modulation

ν	H^0	H^1	H^2	Coding gain at BER of 10^{-5} (dB)		
				NEW	UG	P&A
3	17	02	-	3.0	2.9	2.9
4	33	02	-	3.2	3.1	3.1
5	73	24	16	3.4	3.4	3.4
6	121	044	022	3.6	3.6	3.6
7	355	204	016	3.9	3.9	3.9
8	661	126	354	4.1	4.1	4.1
9	1725	0222	0624	4.3	4.3	4.3
10	3447	0752	0216	4.5	4.5	4.5
11	4451	3062	1724	4.6	-	4.5
12	10013	07476	03014	4.8	-	4.8
13	26135	11354	01706	5.0	-	5.0
14	54077	32206	06574	5.1	-	-
15	142573	021504	036152	5.2	-	5.1
16	277005	116176	035760	5.3	-	-
17	674241	174116	041642	5.3	-	5.3
18	1340765	0574302	0372662	5.3	-	-
19	2514251	1057362	0236376	5.3	-	-
20	4254573	0614572	2671042	5.3	-	-

Table 5.2: Trellis codes for 16-QAM modulation

v	H^0	H^1	H^2	H^3	Coding gain at BER of 10^{-5} (dB)	
					NEW	UG
3	13	02	–	–	3.1	3.1
4	21	04	12	–	3.3	3.3
5	51	32	10	–	3.7	3.7
6	131	062	032	–	4.0	3.9
7	323	056	144	–	4.2	4.2
8	471	310	116	–	4.5	4.5
9	1523	0532	0054	–	4.8	4.7
10	2731	1414	1200	–	4.8	4.8
11	4267	2376	0140	–	5.1	–
12	10365	06014	01024	–	5.2	–
13	36467	06304	14340	–	5.3	–
14	53531	20242	24714	–	5.4	–
15	162515	057022	062276	–	5.6	–
16	242123	165172	064140	–	5.7	–
17	627325	275752	007350	–	5.7	–
18	1161671	0431510	0374242	0500636	5.7	–
19	3122511	1165602	0741314	0170622	5.7	–
20	5110025	3637202	2242414	0727014	5.7	–

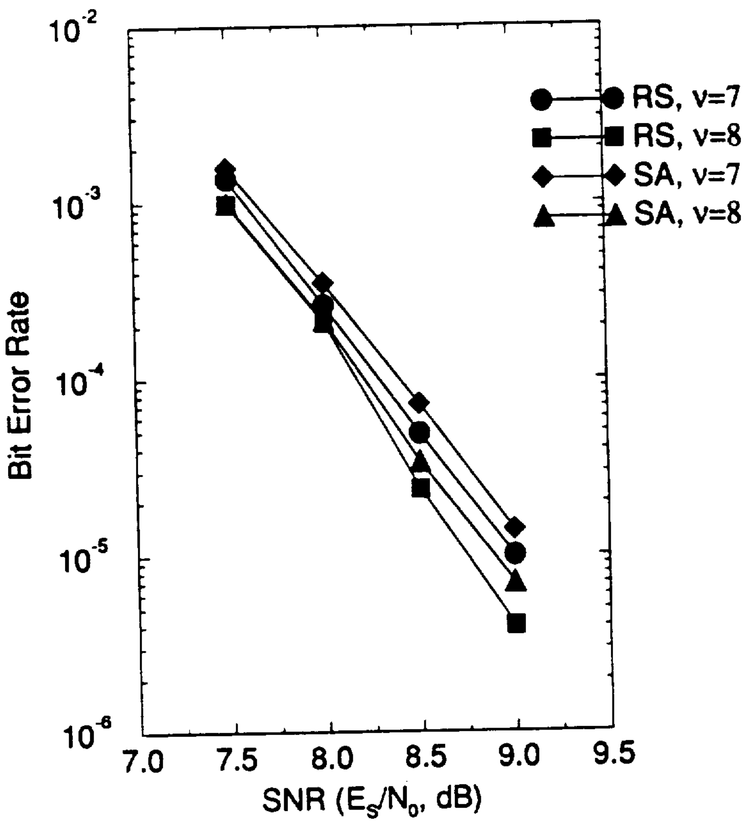


Figure 5.3: Comparison of two code construction algorithms

the SNR's required to achieve a certain BER for a coded system and an uncoded system. They were determined by simulations. It is amazing to note that the new codes achieve about the same or slightly larger real coding gains than the best known codes. At lower SNR's (smaller BER), the new codes perform even better.

Tables 5.1 and 5.2 show that a trellis code with constraint length $\nu = 16$ can achieve the channel cut-off rate at a BER of 10^{-5}. The real coding gains at a BER of 10^{-5} remain about the same for longer codes. Actually, the improvement of the gains are not noticeable at a BER of 10^{-5} when the cut-off bound is achieved. But the performance indeed continues to improve with the increase of ν. At lower BER's, the coding gains grow with the ν for ν larger than 16. For example, codes with a constraint length $\nu = 18$ can achieve a BER of about 10^{-6} at the cut-off rate bound.

This shows a real coding gain of 6.2 and 6.6 dB over an uncoded 4-PSK and 8-QAM system respectively.

The performance of rate 2/3 trellis codes for 8-PSK modulation using Ungerboeck codes, Porath and Aulin codes, and the new codes is further compared using sequential decoding at an SNR= 7.75 dB. The results is shown in Figure 5.4. It shows that the

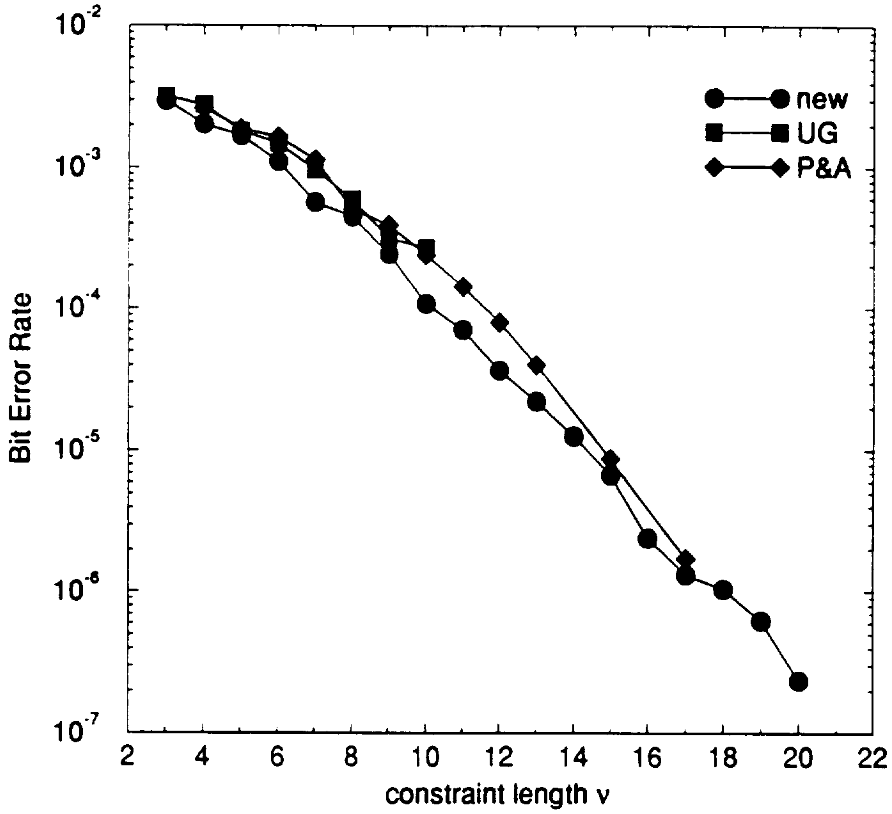


Figure 5.4: Performance comparison of trellis coded 8-PSK codes

new codes have the best performance over the entire range of constraint lengths. Similar comparison for rate 3/4 trellis codes for 16-QAM modulation is shown in Figure 5.5.

To see how the new codes perform over a wide range of SNR's, trellis codes for 8-PSK modulation with $\nu = 4$ and $\nu = 7$ are decoded using the Viterbi algorithm

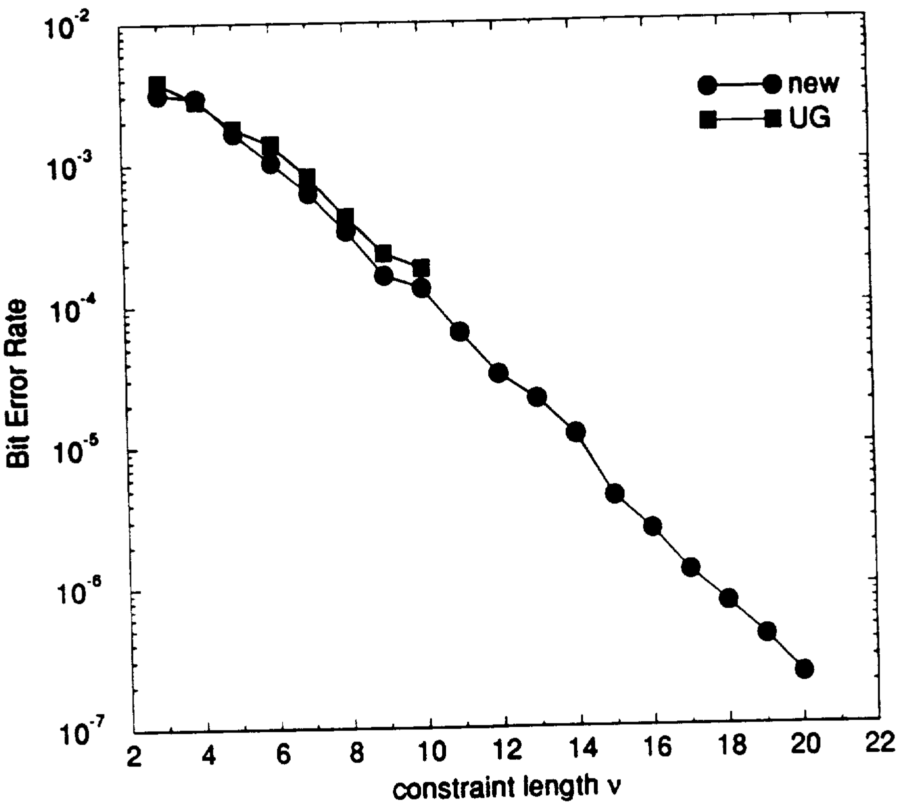


Figure 5.5: Performance comparison of trellis coded 16-QAM codes

and sequential decoding. The performance of the new codes along with Ungerboeck codes of the same constraint length is shown in Figure 5.6 and 5.7 respectively. Figure 5.6 shows that at low SNR, the new codes perform slightly better than the Ungerboeck codes. This is due to the fact that the Ungerboeck codes have larger path multiplicities than the new codes. A calculation of the distance spectrum shows that in many cases the new codes have smaller multiplicities but less free distance than the Ungerboeck codes. This is because the codes are constructed at a low SNR. Figure 5.7 shows that the new codes perform better than the Ungerboeck codes over a wide range of SNR with sequential decoding. This is due to the fact that the Ungerboeck codes were not designed for use with sequential decoding, i.e., their distance profiles

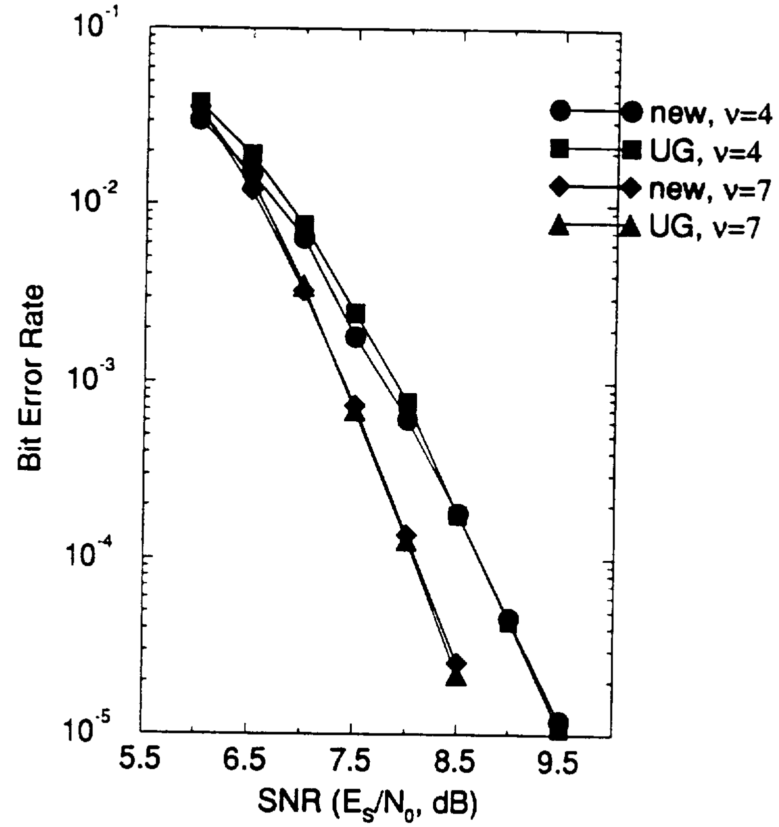


Figure 5.6: Performance of new and Ungerboeck codes with Viterbi decoding

are suboptimum.

5.3 Simulation Results

In the previous section, trellis codes have been constructed using a probabilistic approach. In this section, simulation results are presented to show that the cut-off rate bound can be achieved with the large constraint length trellis codes using sequential decoding.

First, the conventional Fano algorithm is used to decode the trellis coded 8-PSK and trellis coded 16-QAM. In the simulation, a buffer with an infinite size is assumed,

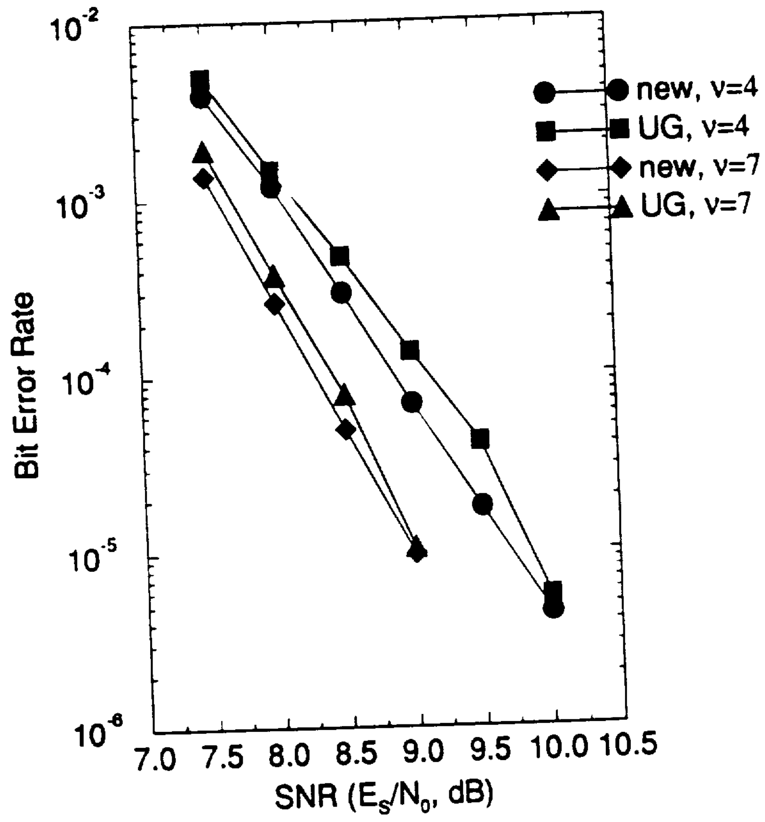


Figure 5.7: Performance of new and Ungerboeck codes with sequential decoding

i.e., the Fano algorithm is allowed to run until the received data are decoded. The simulation results for trellis coded 8-PSK with constraint lengths $\nu = 15$ and $\nu = 16$ along with the cut-off rate bound and the performance of uncoded QPSK ;ystem are shown in Figure 5.8. Figure 5.8 shows that the cut-off rate bound is achieved with constraint length 15 code at a Bit Error Rate (BER) of 10^{-5}. This accounts for about 5.3 dB practical coding gain over an uncoded QPSK system at a BER of 10^{-5}. It also shows that the constraint length $\nu = 16$ code can achieve a BER of 3.0×10^{-6} at the channel cut-off rate bound. Figure 5.4 shows that the performance of trellis coded 8-PSK improves with the code constraint length. We expect that the cut-off rate bound can be achieved using our codes with larger constraint lengths at a BER of

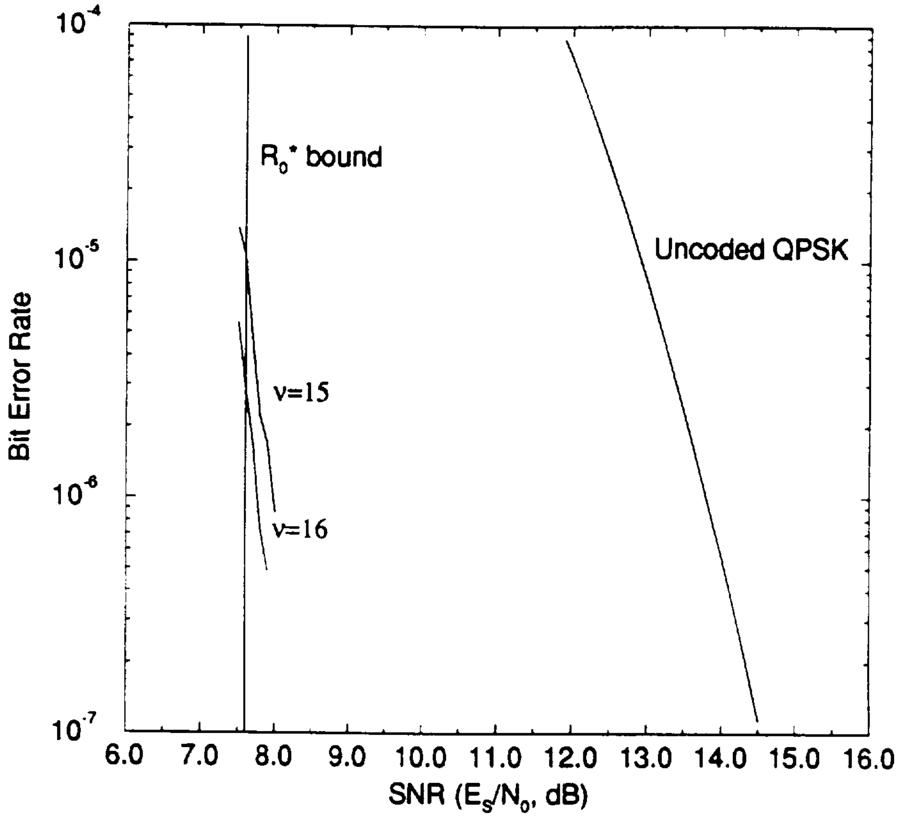


Figure 5.8: Performance of large constraint length trellis coded 8-PSK using sequential decoding

10^{-6} and below. Simulation indicates that the trellis coded 8-PSK with a constraint length of 18 can achieve a BER smaller than 10^{-6} at the cut-off rate bound. Looking at Figure 5.8, we find that the required SNR for an uncoded QPSK to achieve a BER of 10^{-6} is about 13.8 dB. Thus, a trellis coded 8-PSK with a constraint length 18 using sequential decoding can achieve about 6.2 dB real coding gain over an uncoded QPSK system at a BER of 10^{-6}.

Similarly, in Figure 5.9, we show the simulation results for trellis coded 16-QAM with constraint lengths $\nu = 15$ and $\nu = 16$ along with the cut-off rate bound and the performance of uncoded 8-QAM system using the Fano algorithm. It shows that

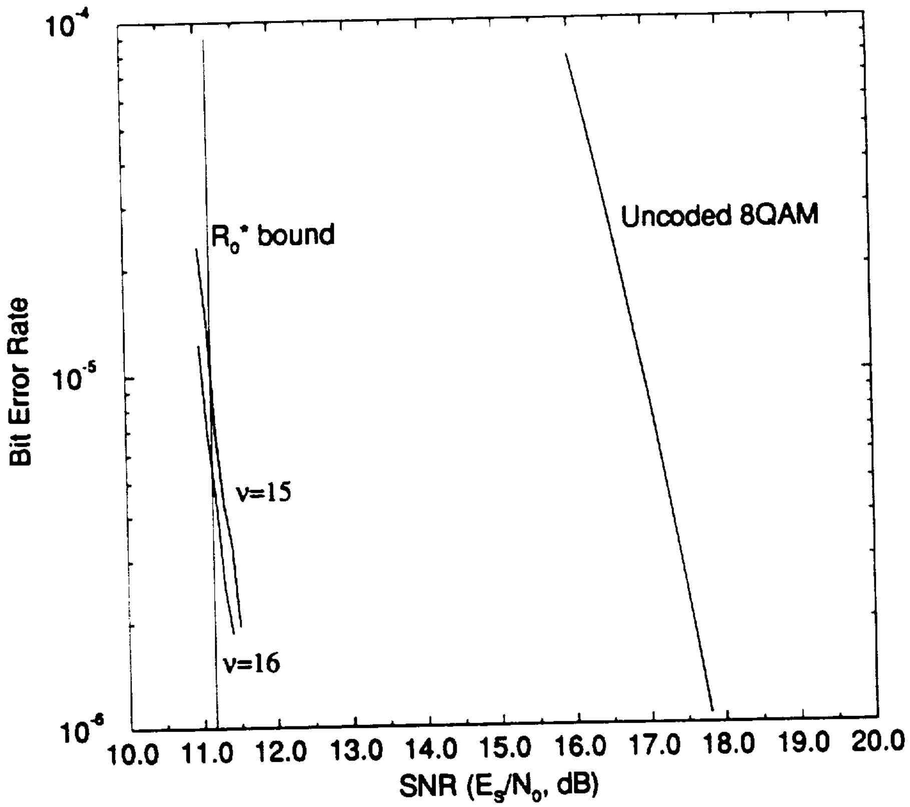


Figure 5.9: Performance of large constraint length trellis coded 16-QAM using sequential decoding

the cut-off rate bound can also be achieved with constraint length 15 code at a Bit Error Rate (BER) of 10^{-5}. This accounts for about 5.7 dB practical coding gain over an uncoded 8-QAM system at a BER of 10^{-5}. Figure 5.9 shows that the constraint length $\nu = 16$ code results in a BER of 6.0×10^{-6} at the channel cut-off rate bound. In Figure 5.5, we see that the performance of trellis coded 16-QAM also improves with the code constraint length. We also note that the trellis coded 16-QAM with a constraint length of 18 can achieve a BER smaller than 10^{-6} at the cut-off rate bound, which accounts for about 6.6 dB real coding gain over an uncoded 8-QAM system at a BER of 10^{-6}.

Figure 5.8 and 5.9 show that the cut-off rate bounds can indeed be achieved. However, in practice, the assumption of an infinite buffer is not realistic. The buffer will always be finite. In this case, the buffer will overflow eventually no matter how large it is since the computational effort is a random variable with a Pareto distribution. When a buffer overflows, the incoming data will be lost in a continuous communication system. The amount of the lost data will depends on the time that the decoder spend to overcome some severely corrupted branches. In Figure 5.10, we show the performance of trellis coded 8-PSK with constraint length $\nu = 15$ and 16 using the BLA described in Chapter 3, which guarantees erasurefree decoding. The block

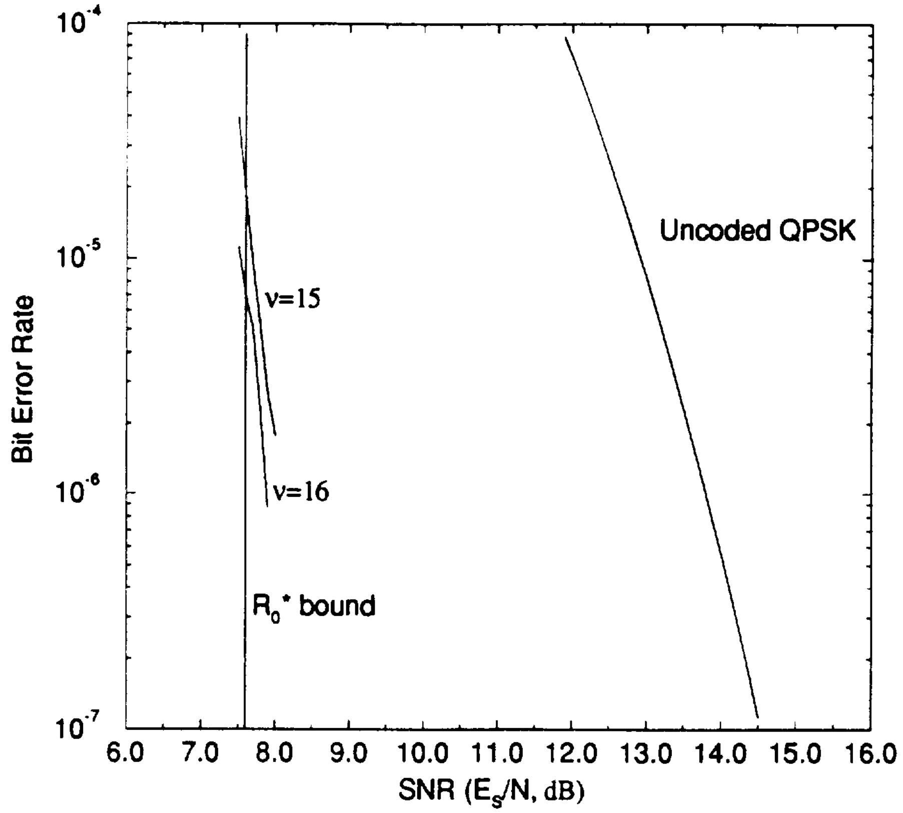


Figure 5.10: Performance of large constraint length trellis coded 8-PSK using the BLA

decoding BLA with a speed factor $\mu = 16$, a block size $L = 512$ branches (signals), and a buffer size $B = 64$ K branches (signals) was used for our simulations. It shows that the cut-off rate bound can be achieved at a BER of 10^{-5} with a $\nu = 16$ code, one more than the case of conventional sequential decoding as shown in Figure 5.8. Figure 5.11 shows the performance of trellis coded 16-QAM with constraint length

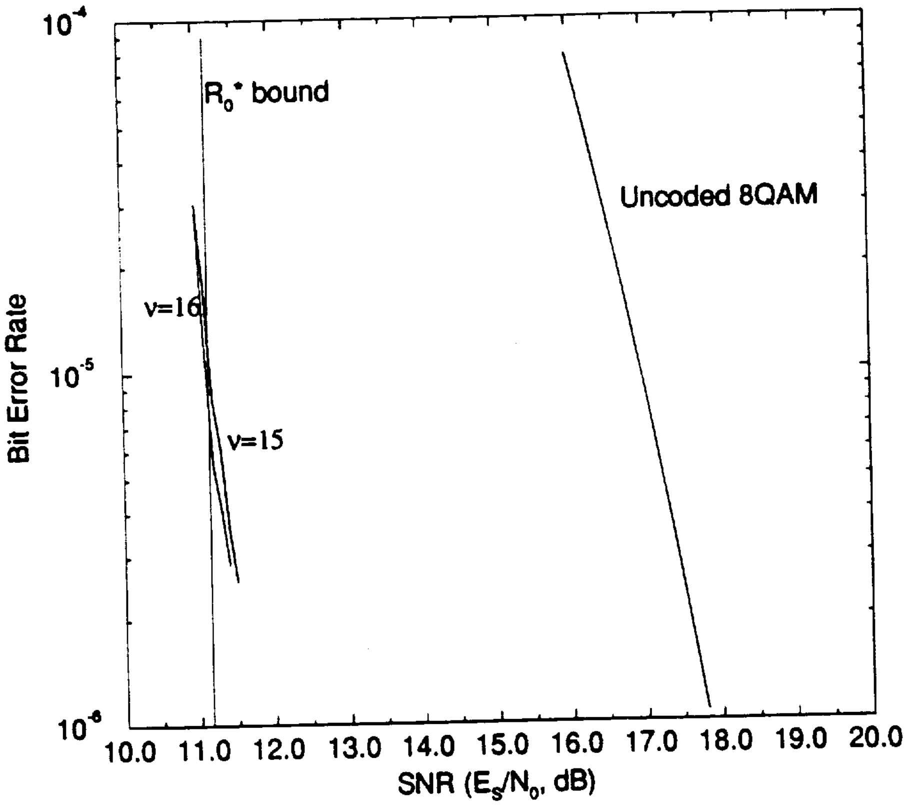


Figure 5.11: Performance of large constraint length trellis coded 16-QAM using the BLA

$\nu = 15$ and 16 using the BLA with $\mu = 16$, $L = 512$ branches (signals), and $B = 64$ K branches. Similar conclusion can be drawn from Figure 5.11, i.e., the cut-off rate bound can be achieved at a BER of 10^{-5} with a $\nu = 16$ code. Using similar arguments as in the case of conventional sequential decoding, we see that the coded 8-PSK and

16-QAM can achieve about 5.3 dB and 5.7 dB real coding gains over uncoded QPSK and 8-QAM modulations. Our simulations indicate that the trellis coded 8-PSK and 16-QAM with a constraint length of 19 can achieve a BER of smaller than 10^{-6} at the cut-off rate bounds. This accounts for about 6.2 dB and 6.6 dB real coding gains over uncoded QPSK and 8-QAM modulations at a BER of 10^{-6}.

6

SHAPING AND CODING

So far, we have discussed the application of sequential decoding to trellis codes and the construction of trellis codes for use with sequential decoding. It has been shown that the cut-off rate bound R_0^* can be achieved. In Chapter 1, it was shown that the difference between R_0 and R_0^* is about 1.5 dB for higher spectral efficiencies. This was defined as shaping gain with respect to channel cut-off rate.

In a coded modulation system, a shaping gain can be achieved by using either higher dimensional spherical constellations[5, 22, 27] or appropriately designed shaping codes [3, 4, 24, 48]. However, it has been recognized [4, 24] that it is advantageous to pursue shaping gain directly via a shaping code rather than indirectly via shaping a higher dimensional constellation. Existing schemes[4, 24] that employ shaping and coding utilizes one or more normal codes and a shaping code separately. Forney and Wei [24, 27] assert that sıaping and coding are separable and additive at high data rates (spectral efficiencies). However, Pottie and Calderbank [62] recently argued that shaping and coding may not be separable in the limit of large code complexity. Are they talking about the same thing? What is the separability of shaping and coding? What does it imply? In this chapter, we try to answer these questions. In Section 6.1, coded modulation is reviewed. In Section 6.2, shaping gain is defined in a shaped modulation system. In Section 6.3, the separability of shaping and coding in a coded/shaped system is examined.

6.1 Coded Modulation

Only QAM modulation is considered in this chapter. Coded modulation combines coding and modulation into one scheme. It has been shown that significant coding gain can be achieved by doing so. To transmit k_c information bits/T (T is the modulation time period) using a rate k_c/n_c code, a 2^{n_c} point constellation is needed. The coding Constellation Expansion Ratio is defined as

$$CER_c = 2^{n_c - k_c}. \tag{6.1}$$

Normally, the points in a coded modulation scheme are used with equal probability. Assume that the minimum (squared) distance between the points is d_0^2. Then, the average energy per signal can be obtained using a continuous approximation[27] as

$$E_c = 2^{n_c} d_0^2/6. \tag{6.2}$$

Assume that the minimum distance in the uncoded modulation system is d_u^2. Similarly, the average signal energy for the uncoded system can be obtained as

$$E_u = 2^{k_c} d_u^2/6. \tag{6.3}$$

Suppose that a coded modulation system has a free distance of d_{free}^2. For an uncoded modulation system to achieve the same performance as the coded modulation system, the minimum distance of the uncoded modulation system d_u^2 must be as large as d_{free}^2, i.e., an uncoded system requires an average signal energy of

$$E_u = 2^{k_c} d_{free}^2/6 \tag{6.4}$$

to maintain the same performance. The coding gain G_c can then be defined as the energy reduction of a coded modulation system over an uncoded modulation system expressed in dB, i.e.,

$$G_c = 10\log_{10}\frac{2^{k_c} d_{free}^2}{2^{n_c} d_0^2} = 10\log_{10}\frac{d_{free}^2}{CER_c \times d_0^2}. \tag{6.5}$$

The merit of coded modulation may also be demonstrated by random coding arguments. Shannon[68] showed that arbitrarily low error probability can be achieved when coding is employed as long as the transmission rate is smaller than the channel capacity. The channel capacity for a 2^{n_c} point constellation is given by[70]

$$C^* = n_c - \frac{1}{2^{n_c}}\sum_{k=0}^{2^{n_c}-1} E_z\left\{\log_2 \sum_{i=0}^{2^{n_c}-1} exp\left[-\frac{|z-a^i|^2 - |z-a^k|^2}{2\sigma^2}\right]\right\}, \tag{6.6}$$

where E_z denotes the expectation of z and $\{a^i, i = 0, 1, \cdots, 2^{n_c}-1\}$ are the constellation points. C^* can be evaluated by Monte Carlo techniques for a given Signal to Noise Ratio (SNR), which is the average signal energy over the single sided noise power spectrum N_0.

Example 1. $SNR = 9.30$ dB is required to achieve $C^* = 3$ bits/T for a 16-QAM modulation.

6.2 Shaped Modulation

We are interested in using a shaping code to achieve shaping gain. In a shaped system, the signa with less energy are used more often and thus the signals are nonequiprobable.

Similar to coded modulation, the constellation is expanded from 2^{r_s} to 2^{n_s} points to transmit r_s bits when a rate r_s/n_s shaping code is used. The shaping Constellation Expansion Ratio is

$$CER_s = 2^{n_s - r_s}. \tag{6.7}$$

Let E_p be the average signal energy of a shaped modulation scheme. Shaping does not change the minimum distance between signal points in the constellation. Thus, the performance remains the same. A shaped modulation scheme can transmit up to

$$H(\mathbf{p}) = -\sum_{i=0}^{2^{n_s}-1} p_i \log_2 p_i \tag{6.8}$$

bits of information, where $\mathbf{p}$ denotes the probability vector $(p_0, p_1, \cdots, p_{2^{n_s}-1})$ and p_i is the probability that the shaping scheme selects the i-th point in the constellation. Then the shaping gain G_s can be defined as the energy reduction of the shaped modulation system over the unshaped modulation system at the same spectral efficiency, i.e.,

$$G_s = 10 \log_{10} \frac{E_u}{E_p}, \tag{6.9}$$

where E_u denotes the average signal energy of the unshaped modulation system, which can be computed using the continuous approximation as

$$E_u = 2^{r_s} d_u^2/6 = 2^{H(p_0, p_1, \cdots, p_{2^{n_s}-1})} d_u^2/6. \tag{6.10}$$

E_u can also be obtained by exact calculation. Suppose that the i-th point in the constellation has a signal energy of E_i, we then have the average signal energy of such a signaling system

$$E(\mathbf{p}) = \sum_{i=0}^{2^{n_s}-1} p_i E_i. \tag{6.11}$$

An ideal shaping scheme is the one that minimizes $E(\mathbf{p})$ with the constraints that $\sum_{i=0}^{2^{n_s}-1} p_i = 1$ and $H(\mathbf{p}) = r_s$. Define

$$f(\mathbf{p}) = r_s - H(\mathbf{p}) \tag{6.12}$$

and

$$g(\mathbf{p}) = \sum_{i=0}^{2^{n_s}-1} p_i - 1. \tag{6.13}$$

We need to minimize $E(\mathbf{p})$ subject to the constraints $f(\mathbf{p}) = 0$ and $g(\mathbf{p}) = 0$ to obtain E_p. We define

$$F(\mathbf{p}, \lambda, \phi) = E(\mathbf{p}) + \lambda f(\mathbf{p}) + \phi g(\mathbf{p}). \tag{6.14}$$

Applying Lagrange multipliers, we obtain

$$\begin{aligned} E_i + \lambda(\log_2 p_i + \log_2 e) + \phi = 0, \quad i = 0, 1, \cdots, 2^{n_s} - 1, \\ \textstyle\sum_{i=0}^{2^{n_s}-1} p_i - 1 = 0, \\ r_s - H(\mathbf{p}) = 0, \end{aligned} \tag{6.15}$$

i.e., p_i should be chosen as

$$p_i = 2^{-\frac{\phi+E_i}{\lambda}-\log_2 e}, \tag{6.16}$$

where λ and ϕ are chosen such that the probabilities sum to 1 and the entropy $H(\mathbf{p})$ is equal to the desired transmission rate r_s. (6.16) shows a Gaussian-like distribution, agreeing with the intuition of Forney and Wei[27].

Example 2. Shaped 64-QAM over 8-QAM (regular). 64-QAM is shown in Figure 6.1. The base (unshaped) modulation constellation is shown in Figure 6.2. Suppose

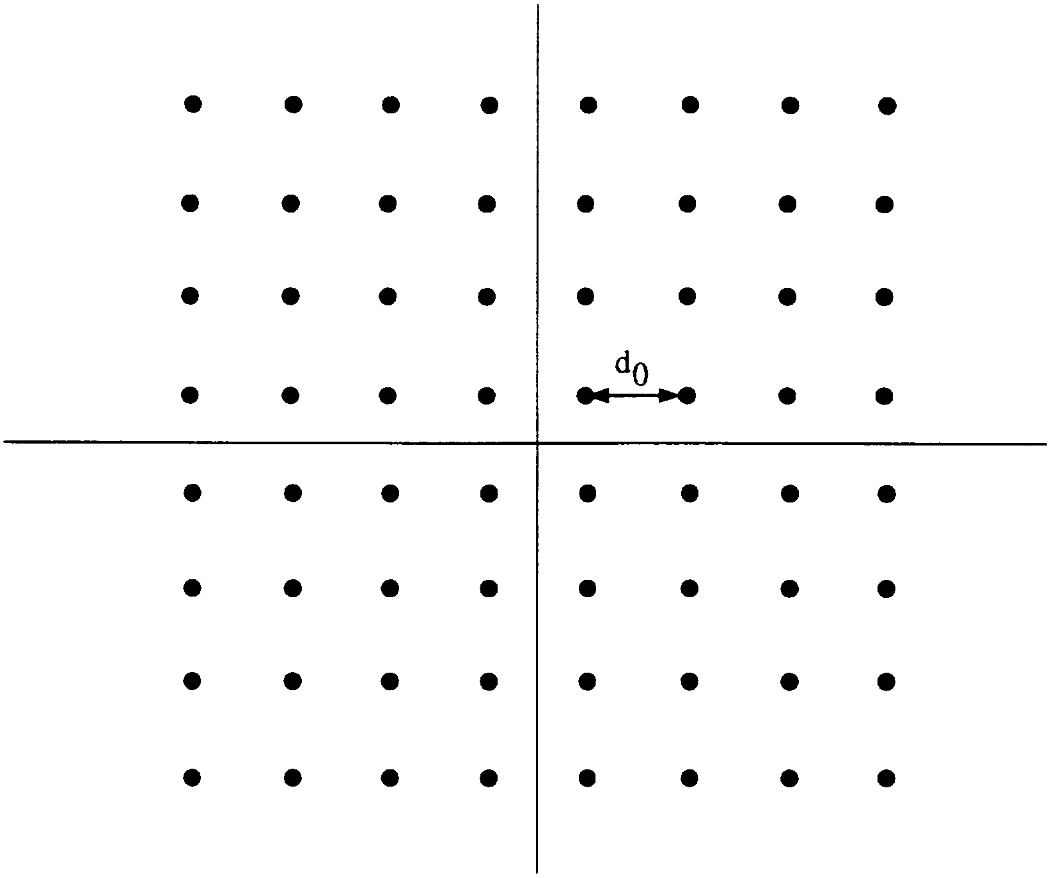


Figure 6.1: Constellation of 64-QAM modulation

that $d_u^2 = 4$ and let $d_0 = d_u$. Then we obtain $E_u = 5$ using exact calculation. Using (6.16), we obtain a set of p such that $E_p = 3.75$. Thus, $G_s = 1.25$ dB.

Example 3. Shaped 64-QAM over 8-QAM (non-regular). The base (unshaped) modulation constellation is shown in Figure 6.3. Suppose that $d_u^2 = 4$ and let $d_0 = d_u$. Then we obtain $E_u = 6$ using exact calculation. Using (6.16), we obtain a set of p

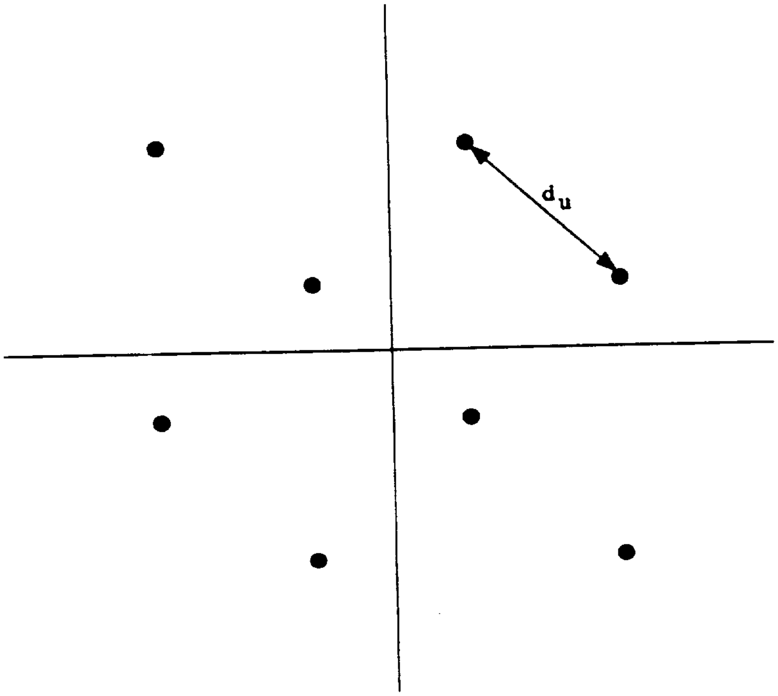


Figure 6.2: Constellation of regular 8-QAM modulation

such that $E_p = 3.75$. Thus, $G_s = 2.04$ dB.

Example 4. Shaped 64-QAM over 16-QAM. T: base (unshaped) modulation constellation is shown in Figure 6.4. Suppose that $d_u^2 = 4$ and let $d_0 = d_u$. Then we obtain $E_u = 10$ using exact calculation. Using (6.16), we obtain a set of p such that $E_p = 7.5$. Thus, $G_s = 1.25$ dB.

Existing shaping schemes [3, 4, 24, 48] can achieve a good portion of this shaping gain.

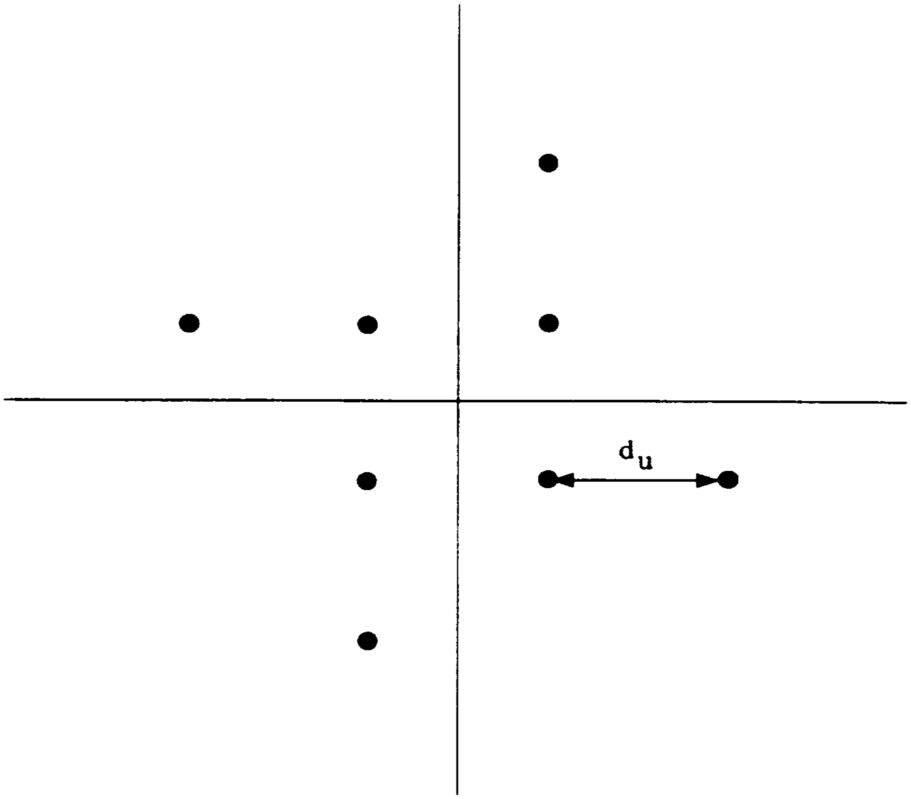


Figure 6.3: Constellation of non-regular 8-QAM modulation

6.3 Coded/Shaped Modulation

There are two ways to integrate coding and shaping in a modulation system. Figure 6.5 shows a separated coded/shaped modulation system in a parallel structure. Forney's trellis coded/shaped scheme[24] and Calderbank and Ozarow's multilevel coded/shaped scheme[4] can be represented this way. Calderbank and Ozarow[4] have stated that "the separability of coding and shaping means that one part of the input data stream drives C_1, C_2 (normal codes) and produces coding gain (over un-

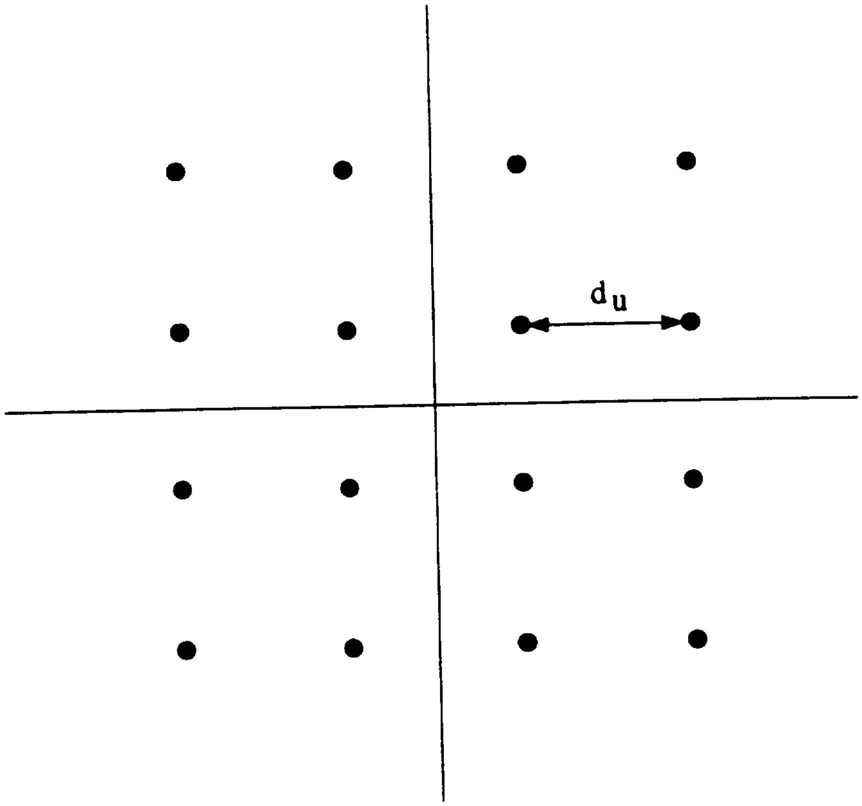


Figure 6.4: Constellation of 16-QAM modulation

coded transmission), and a different part of the input data stream drives C_3 (shaping code) and produces shaping gain (over equiprobable signaling). The two types of gain add." This definition of separability was motivated by the separated structure as shown in Figure 6.5. Obviously, schemes with this structure satisfy the first condition of the separability definition. Do they satisfy the second condition, i.e., do shaping gain and coding gain add? Furthermore, can schemes with this structure achieve Shannon's bound[68]? We now try to answer these questions.

Let E_{cp} be the average signal energy in a coded/shaped modulation system. We

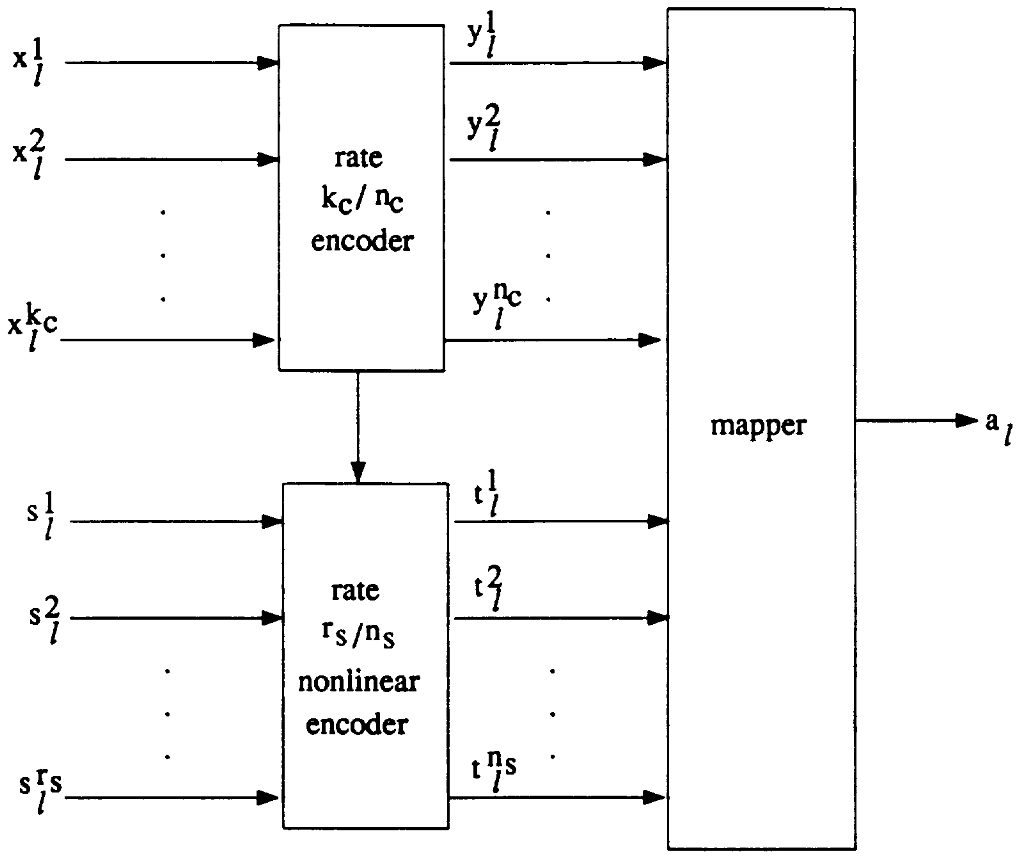


Figure 6.5: A separated coded/shaped system in parallel structure

define the total gain as

$$\begin{aligned} G &= 10\log_{10}\frac{E_u}{E_{cp}} \\ &= 10\log_{10}\frac{E_u}{E_c} + 10\log_{10}\frac{E_c}{E_{cp}} \\ &= G_c + G_s', \end{aligned} \tag{6.17}$$

where G_s' is the shaping gain of a coded/shaped modulation system over a coded modulation system. If $G'_s = G_s$, then shaping gain and coding gain are additive as shown in (6.17). We now give a counter example showing that $G'_s \neq G_s$.

Example 5. Examples 2 and 4 show that G_s of 64-QAM over 8-QAM or 16-QAM is 1.25 dB. Example 1 says that $SNR = 9.30$ dB is required to achieve $C^* = 3.0$ bits/T. Then, if $G'_s = G_s$, we only need

$$\begin{aligned} SNR &= 10\log_{10}\frac{E_{sp}}{N_0} \\ &= 10\log_{10}\frac{E_c}{N_0} + 10\log_{10}\frac{E_{sp}}{E_c} \\ &= 9.30 - G'_s \\ &= 8.05\ dB \end{aligned} \tag{6.18}$$

to transmit 3.0 bits of information per signal when shaping is employed. This contradicts Shannon's bound[68], which says that we need at least $SNR = 8.45$ dB to realize reliable communication at a transmission rate of 3 bits/symbol with two dimensional signals. We conclude that the second condition of Calderbank and Ozarow's definition of separability cannot be satisfied. This can be attributed to the fact that the signals in a coded sequence are no longer independent.

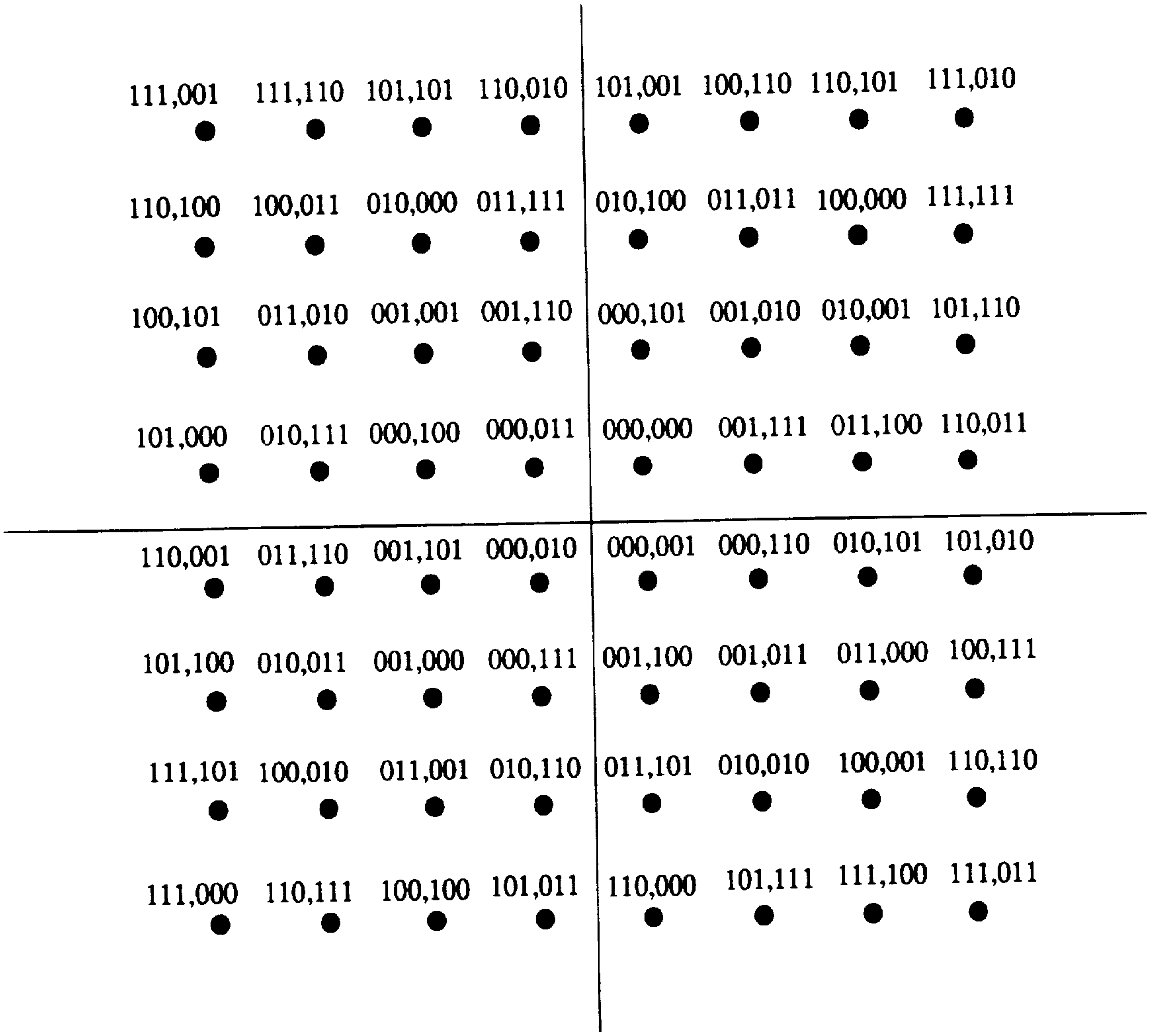
Examining the above example, we see that the contradiction arises because the shaping gain γ_u of an infinite QAM signal set with nonequiprobable signaling (a discrete distribution is applied in the example) over the original QAM signal set with equiprobable signaling is larger than the gap γ_c between the SNR computed using the Shannon's bound (corresponding to nonequiprobable signaling) and the SNR computed using Ungerboeck's formula (corresponding to equiprobable signaling) at the same transmission rate. Noting that both γ_u and γ_c increase with the transmission rate and approach the ultimate shaping gain of 1.53 dB, our example implies that γ_u is always larger than γ_c. Thus, the second condition of Calderbank and Ozarow's definition of separability can only be satisfied in the limit of infinite spectral efficiency.

Now, we return to the second question posed at the beginning of this section, i.e., can schemes with the structure shown in Figure 6.5 achieve Shannon's bound? For a coded/shaped modulation system with the structure shown in Figure 6.5, a $2^{n_c+n_s}$ point signal set is needed and the signal set must be partitioned into 2^{n_c} subsets. Then the free distance in this structure is limited to the minimum distance between points in the subsets $d_{n_c}^2$, i.e., $d_{free}^2 \leq d_{n_c}^2 = a_m d_0^2$, where $a_m = d_{n_c}^2 / d_0^2 < \infty$ since $d_{n_c}^2 < \infty$. Thus, the total gain of such a system is limited, i.e.,

$$\begin{aligned} G &= G_c + G_s' \\ &\leq 10\log_{10}\frac{d_{free}^2}{CER_c \times d_0^2} + 1.53 \\ &\leq 10\log_{10}\frac{a_m}{CER_c} + 1.53 \\ &< \infty. \end{aligned} \tag{6.19}$$

On the other hand, it is shown in [65] that d_{free}^2 increases without bound. Thus, $G_c \to \infty$ and so $G \to \infty$ for a general coded/shaped modulation system. This shows that the Shannon bound cannot be achieved with this structure. It is in this sense that Pottie and Calderbank[62] argued that shaping and coding cannot be separated.

Example 6. Assume that a rate 1/3 shaping code and a rate 2/3 normal code are used to transmit 3 bits/T in a coded/shaped modulation system. Then $CER_c = 2$ and a 64-QAM constellation is required. The 64-QAM signal set must be partitioned into 8 subconstellations. The shaping code selects one of the subconstellations and the normal code selects a point in a subconstellation. Figure 6.6 shows a mapping obtained by set partitioning. It clearly shows that $a_m = 8$ in this case. Thus, the total gain of this system is



111,001	111,110	101,101	110,010	101,001	100,110	110,101	111,010
110,100	100,011	010,000	011,111	010,100	011,011	100,000	111,111
100,101	011,010	001,001	001,110	000,101	001,010	010,001	101,110
101,000	010,111	000,100	000,011	000,000	001,111	011,100	110,011
110,001	011,110	001,101	000,010	000,001	000,110	010,101	101,010
101,100	010,011	001,000	000,111	001,100	001,011	011,000	100,111
111,101	100,010	011,001	010,110	011,101	010,010	100,001	110,110
111,000	110,111	100,100	101,011	110,000	101,111	111,100	111,011

$\{t^3 t^2 t^1, y^3 y^2 y^1\}$

Figure 6.6: Mapping of 64-QAM in a coded/shaped system

$$\begin{aligned} G &\leq 10\log_{10}\frac{a_m}{CER_c} + 1.53 \\ &= 7.55 \;\; dB. \end{aligned} \tag{6.20}$$

Another way to integrate coding and shaping in a modulation system is shown in Figure 6.7. It achieves shaping gain by augmenting the coded modulation using a shaping code, while the coding gain is achieved with a normal code. Obviously, this structure avoids the limitations on coding gain of the parallel scheme. Using similar arguments, we can show that shaping gain and coding gain with this structure still do not add. Another question related to this structure is whether the existing shaping schemes [3, 4, 24, 48] can be adapted to this structure? Our attempts thus far have been unsuccessful. An interesting open question is how to design a shaping scheme that can be integrated with this structure?

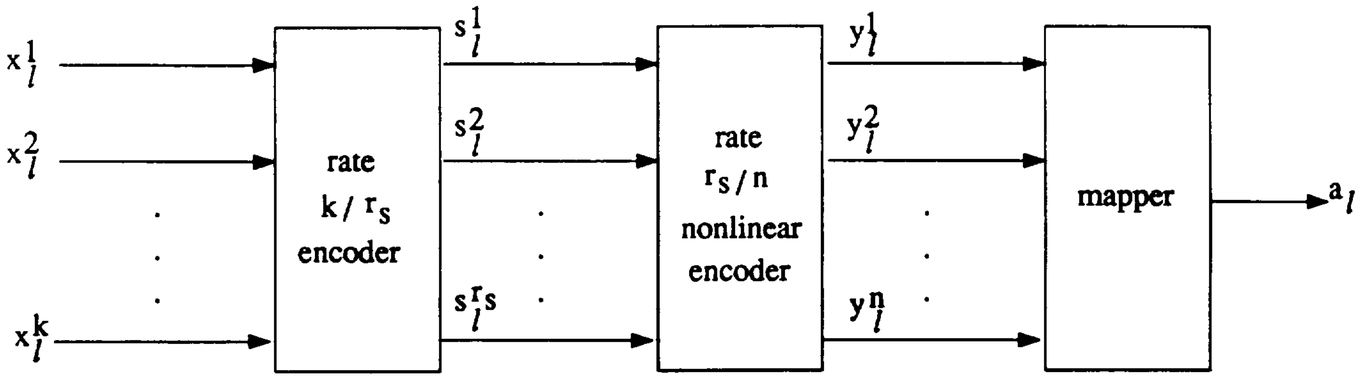


Figure 6.7: A separated coded/shaped system in cascade structure

7

CONCLUSIONS

Channel cut-off rate is considered as the practically achievable rate by many people. In this dissertation, we have investigated approaches to achieve the cut-off rate at a Bit Error Rate (BER) of $10^{-5} - 10^{-6}$ for bandlimited Additive White Gaussian Noise (AWGN) channels. Three aspects of trellis coding have been explored. These are the application of sequetial decoding and its modifications to trellis codes, construction of trellis codes for use with sequetial decoding, and exploration of the relationship between shaping and coding.

In Chapter 2, sequential decoding of trellis codes is addressed. The Fano metric is shown to be a maximum likehood matric for variable length codes on a bandlimited AWGN channel. Demodulator quantization for PSK and QAM modulations is discussed. Rectangular and angular quantization schemes for PSK modulation are compared using simulation. It shows that rectangular quantization scheme outperforms angular scheme at high definitions. A simple method to increase the distance of trellis codes in the tail is presented and the tail's influence on performance is studied. The performance of trellis codes using sequential decoding is then investigated. Simulation results show that sequential decoding performs slightly worse than the Viterbi algorithm for the same constraint length code. However, this suboptimality of sequential decoidng can be overcome using a slightly larger constraint length code with little penalty of computational complexity. It is then shown that the performance of trellis codes using sequential decoding improves steadily with the increase of the code constraint length and the channel cut-off rate bound can be achieved at a BER of 10^{-5}. It is also shown that the distribution of computational effort for sequential decoding of trellis codes can be approximated by a Pareto distribution.

In Chapter 3, an erasurefree sequential decoding algorithm is introduced. Several versions of the algorithm can be obtained by choosing certain parameters and selecting a resynchronization scheme. These can be categorized as block decoding or continuous decoding, depending on the resynchronization scheme. Block decoding is guaranteed to resynchronize at the beginning of each block, but suffers some rate loss when the block length is relatively short. The performance of a typical block decoding scheme is analyzed and we show that significant coding gains over Viterbi decoding can be achieved with much less computational effort. A resynchronization scheme is proposed for continuous sequential decoding. It is shown by analysis and simulation that continuous sequential decoding using this scheme has a high probability of resynchronizing successfully.

In Chapter 4, The relationship between the distance properties of trellis codes and the compuational effort of sequential decoding is studied and trellis codes for 8-PSK and 16-QAM modulation with Optimum Distance Profile (ODP) and Optimum Free Distance (OFD) are constructed. The design criteria for trellis codes with sequential decoding are examined. A comparision of trellis codes with ODP and OFD reveals that both ODP and OFD trellis codes for some constraint lengths may not result in the best trade-off between error performance and computational performance when sequential decoding is used. A new code construction algorithm is proposed to construct robustly good trellis codes for use with sequential decoding. Trellis codes with asymptotic coding gains up to 6.66 dB are obtained using this algorithm and the new codes achieve nearly the same free distances as the OFD codes and nearly the same distance profiles as the ODP codes. Simulation results show that the new codes outperform the best known trellis codes in terms of error probability as well as computational effort.

In Chapter 5, probabilistic construction algorithms are investigated for constructing good long trellis codes that can achieve the channel cut-off rate at a BER of $10^{-5} - 10^{-6}$. The algorithms are motivated by the random coding bound for trellis-type codes. One algorithm begins by choosing a relatively small set of codes randomly. The error performance of each of these codes is evaluated using sequential decoding and the code with the best performance among the chosen set is retained. Another algorithm treats the code constrcution as a combinatorial optimization problem and introduces simulated annealing algorithm to conduct the code search work. Codes for 8-PSK and 16-QAM modulations with constraint lengths ν up to 20 and practical coding gains up to 6.6 dB at a BER of $10^{-5} - 10^{-6}$ are obtained. It is surprising to find out that the new codes found in this paper, which come from a very small set of codes compared to the total number of possible codes, perform about as well as the best known codes at a BER of 10^{-5}. Simulation results show that the codes constructed in this approach can achieve the cut off rate bound at a BER of $10^{-5} - 10^{-6}$ which correspond to 5.3 – 6.6 dB real coding gains over uncoded systems.

In Chapter 6, the separability of shaping and coding in a coded/shaped modulation system is examined. It is shown that the existing schemes that employ shaping as well as coding cannot approach Shannon's bound. It is also shown that shaping gain and coding gain do not add in a separated coded/shaped modulation system, i.e., the second condition of Calderbank and Ozarow's definition of separability of shaping and coding (additivity of shaping gain and coding gain) is not satisfied. This can be attributed to the fact that the signals in a coded sequence are no longer independent.

BIBLIOGRAPHY

[1] T. Berger, *Rate Distortion Theory* , Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[2] E. Biglieri and P. J. McLane, "Uniform Distance and Error Probability Properties of TCM Schemes," *IEEE Trans. Commun.*, **COM–39**, pp. 41–53, January 1991.

[3] A. R. Calderbank and M. Klimesh, "Balanced Codes and Nonequiprobable Signaling," *IEEE Trans. Inform. Theory,* **IT–38**, pp. 1119–1122, May 1992.

[4] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable Signaling on the Gaussian Channel," *IEEE Trans. Inform. Theory,* **IT–36**, pp. 726–740, July 1990.

[5] A. R. Calderbank and N. J. A. Sloane, "New Trellis Codes Based on Lattices and Cosets," *IEEE Trans. Inform. Theory,* **IT–33**, pp. 177–195, March 1987.

[6] P. R. Chevillat and D. J. Costello,Jr., "Distance and Computation in Sequential Decoding," *IEEE Trans. Commun.*, **COM–24**, pp. 440–447, April 1976.

[7] P. R. Chevillat and D. J. Costello,Jr., "A Multiple Stack Algorithm for Erasurefree Decoding of Convolutional Codes," *IEEE Trans. Commun.*, **COM–25**, pp. 1460–1470, December 1977.

[8] P. R. Chevillat and D. J. Costello,Jr., "An Analysis of Sequential Decoding for Specific Time-Invariant Convolutional Codes," *IEEE Trans. Inform. Theory,* **Vol. IT–24**, pp. 443–451, July 1978.

[9] G. C. Clark and J. B. Cain, *Error-Correcting Codes for Digital Communications,* New York: Plenum, 1981.

[10] D. J. Costello,Jr., "A Construction Technique for Random-Error-Correcting Convolutional Codes," *IEEE Trans. Inform. Theory,* **IT–15**, pp.631–636 , September 1969.

[11] D. J. Costello, Jr., L. C. Perez, and F. Q. Wang, "Bandwidth Efficient CCSDS Coding Standard Proposals", *Semi-Annual Status Report*, NASA Grant NAG 5-557, Department of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, May 1992.

[12] A. Drukarev and D. J. Costello,Jr., "Hybrid ARQ Error Control Using Sequential Decoding," *IEEE Trans. Inform. Theory,* **IT–29**, pp. 521–535, July 1983.

[13] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using Simulated Annealing to Design Good Codes," *IEEE Trans. Inform. Theory*, **IT–33**, pp. 116–123, January 1987.

[14] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Trans. Inform. Theory*, **IT–9**, pp. 64–74, April 1963.

[15] N. Farvardin, "A Study of Vector Quantization for Noisy Channel," *IEEE Trans. Inform. Theory*, **IT–36**, pp. 799–809, July 1990.

[16] G. D. Forney, Jr., "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inform. Theory*, **IT–16**, pp. 720–738, November 1970.

[17] G. D. Forney, Jr., "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Inform. Theory*, **IT–18**, pp. 363–378, May 1972.

[18] G. D. Forney, Jr., "The Viterbi Algorithm," *Proc. IEEE*, **61**, pp. 268–278, March 1973.

[19] G. D. Forney, Jr., "Convolutional Codes II: Maximum – Likelihood Decoding," *Inform. and Control*, **Vol. 25**, pp. 222–266, July 1974.

[20] G. D. Forney, Jr., "Coset Codes I: Introduction and Geometrical Classification," *IEEE Trans. Inform. Theory*, **IT–34**, pp. 1123–1151, September 1988.

[21] G. D. Forney, Jr., "Coset Codes II: Binary Lattices and Related Codes," *IEEE Trans. Inform. Theory*, **IT–34**, pp. 1152–1187, September 1988.

[22] G. D. Forney, Jr., "Multidimensional Constellations - Part II: Voronoi Constellations," *IEEE J. Sel. Areas Commun.*, **SAC–7**, pp. 941–958, August 1989.

[23] G. D. Forney, Jr., "Geometrically Uniform Codes," *IEEE Trans. Inform. Theory*, **IT–37**, pp. 1241–1260, September 1991.

[24] G. D. Forney, Jr., "Trellis Shaping," *IEEE Trans. Inform. Theory*, **IT–38**, pp. 281–300, March 1992.

[25] G. D. Forney, Jr. and E. K. Bower, "A High Speed Sequential Decoder: Prototype Design and Test ," *IEEE Trans. Commun. Technol.*, **COM–19**, pp. 821–835, October 1971.

[26] G. D. Forney, Jr., R. G. Gallager, G. R. Lang, F. M. Longstaff, and S. U. Qureshi, Efficient Modulation for Band-limited Channels," *IEEE J. Sel. Areas Commun.*. **SAC–** , pp. 632–647, Sept. 1984.

[27] G. D. Forney, Jr. and L. F. Wei, "Multidimensional Constellations I: Introduction, Figures of Merit, and Generalized Cross Constellation," *IEEE J. Sel. Areas Commun.*, **SAC–7**, pp. 877–892, August 1989.

[28] R. G. Gallager, *Information Theory and Reliable Communication*, New York, Wiley, 1968.

[29] J. M. Geist, "An Emprical Comparison of Two Sequential Decoding Algorithms," *IEEE Trans. Commun. Technol.*, **COM–19**, pp. 415–419, August 1971.

[30] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston, Kluwer, 1992.

[31] K. S. Gilhousen, J. A. Heller, I. M. Jacobs, and A. J. Viterbi, "Coding Study for High Data Rate Telemetry Links", Linkabit Corp. NASA CR–114278 Contract NAS 2–6024, 1971.

[32] D. Haccoun and M. Ferguson, "Generalized Stack Algorithm for Decoding Convolutional Codes," *IEEE Trans. Inform. Theory*, **IT–21**, pp. 638–651, November 1975.

[33] J. Hagenauer, "High Rate Convolutional Codes with Good Distance Profiles," *IEEE Trans. Inform. Theory*, **IT–23**, pp.615–618 , September 1977.

[34] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and Its Applications," *Proc. GLOBECOM'89*, Dallas, TX, pp. 47.1.1.–47.1.7., November 1989.

[35] T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," *IEEE Trans. Inform. Theory*, **IT–33**, pp. 866–876, November 1987.

[36] H. Imai and S. Hirakawa, "A New Multilevel Coding Method Using Error Correcting Codes," *IEEE Trans. Inform. Theory*, **IT–23**, pp. 371–377, May 1977.

[37] I. M. Jacobs and E. R. Berlekamp, "A Lower Bound to the Distribution of Computation for Sequential Decoding," *IEEE Trans. Inform. Theory*, **IT–13**, pp. 167–174, April 1967.

[38] F. Jelinek, "A Fast Sequential Decoding Algorithm Using a Stack," *IBM J. Res. Develop.*, **Vol. 13**, pp. 675–685, November 1969.

[39] R. Johannesson, "Robustly-Optimal Rate One-Half Binary Convolutional Codes," *IEEE Trans. Inform. Theory*, **IT–21**, pp. 464–468, July 1975.

[40] R. Johannesson, "The Error Probability of General Trellis Codes with Applications to Sequential Decoding," *IEEE Trans. Inform. Theory*, **IT–23**, pp. 609–611, September 1977.

[41] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Multilevel Block Modulation Codes", *IEEE Trans. Inform. Theory*, **IT–37**, pp. 965–975, July 1991.

[42] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, **vol.–220**, pp. 671–680, May 1983.

[43] J. W. Layland and W. A. Lusbaugh, "A Flexible High Speed Sequential Decoder for Deep Space Channels," *IEEE Trans. Commun. Technol.*, **COM–19**, pp. 813–820, October 1971.

[44] L. N. Lee, "On Optimal Soft-Decision Demodulation," *IEEE Trans. Inform. Theory*, **IT–22**, pp. 437–444, July 1976.

[45] C. F. Lin and J. B. Anderson, "M-algorithm Decoding of Channel Convolutional Codes," in *Conf. Record , Princeton Conf. Inform. Sci. Syst., Princeton, NJ*, pp. 362–365, March 1986.

[46] S. Lin and D. J. Costello,Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, New Jersey, 1983.

[47] S. Lin and H. Lyne, "Some Results on Binary Convolutional Code Generators," *IEEE Trans. Inform. Theory*, **IT–13**, pp. 134–139, January 1967.

[48] J. N. Livingston, "Shaping Using Variable-Size Regions," *IEEE Trans. Inform. Theory*, **IT–38**, pp. 1347–1353, July 1992.

[49] S. S. Malladi, F. Q. Wang, D. J. Costello, Jr., and H. C. Ferreira, "Construction of Trellis Codes with a Good Distance Profile", *IEEE Trans. Commun.*, to appear.

[50] J. L. Massey, *Threshold Decoding*, MIT Press, Cambridge, Mass., 1963.

[51] J. L. Massey, "Variable-Length Codes and the Fano Metric," *IEEE Trans. Inform. Theory*, **IT–18**, pp. 196–198, January 1972.

[52] J. L. Massey, "Coding and Modulation in Digital Communications," *Proc. 1974 Int. Zurich Seminar Digital Commun.*, Zurich, Switzerland, pp. E2(1)-(4), March 1974.

[53] J. L. Massey and D. J. Costello, Jr., "Nonsystematic Convolutional Codes for Sequential Decoding in Space Applications", *IEEE Trans. Commun. Tech.*, **COM–19**, pp. 806–813, October 1971.

[54] J. L. Massey and M. K. Sain, "Inverses of Linear Sequential Circuits," *IEEE Trans. Comput.*, **C–17**, pp. 330–337, April 1968.

[55] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fasting Computing Machines," *J. Chem. Phys.*, **vol.–21**, June 1953.

[56] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Trans. Inform. Theory*, **IT–15**, pp. 177–179, January 1969.

[57] R. D. Parsons and S. G. Wilson, "Polar Quantization for Coded PSK Transmission," *IEEE Trans. Commun.*, **COM–38**, pp. 1511–1519, September 1990.

[58] S. S. Pietrobon and D. J. Costello, Jr., "Trellis Coding with Multidimensional QAM Signal Sets," submitted to the *IEEE Trans. Inform. Theory*, April 1991.

[59] S. S. Pietrobon, R. H. Deng, A. Lafanechere, G. Ungerboeck, and D. J. Costello, Jr., "Trellis-Coded Multidimensional Phase Modulation," *IEEE Trans. Inform. Theory*, **IT–36**, pp. 63–89, January 1990.

[60] J. Porath, "Algorithms for Converting Convolutional Codes from Feedback to Feedforward Form and Vice Versa," *IEE Electron. Lett.*, **Vol.25**, pp. 1008–1009, July 1989.

[61] J. Porath and T. Aulin, "Algorithmic Construction of Trellis Codes," submitted to the *IEEE Trans. Commun.*, November 1990.

[62] G. J. Pottie and A. R. Calderbank, "Asymptotic Upper Bounds on the Minimum Distance of Trellis Codes", presented at the 1991 Int. Symp. Inform. Theory, Budapest, Hungary, June 1991.

[63] G. J. Pottie and D. P. Taylor, "A Comparison of Reduced Complexity Decoding Algorithms for Trellis Codes," *IEEE J. Sel. Areas Commun.*, **SAC–7**, pp. 1369–1380, December 1989.

[64] I. Richer, "Sequential Decoding with a Small Digital Computer", MIT Lincoln Laboratory Tech. Report No. 491, January 1972.

[65] M. Rouanne and D. J. Costello, Jr., "A Lower Bound on the Minimum Euclidean Distance of Trellis-Coded Modulation Schemes," *IEEE Trans. Inform. Theory*, **IT–34**, pp. 1011–1020, September 1988.

[66] M. Rouanne and D. J. Costello, Jr., "An Algorithm for Computing the Distance Spectrum of Trellis Codes," *IEEE J. Sel. Areas Commun.*, **SAC–16**, pp. 929–940, August 1989.

[67] J. E. Savage, "Sequential Decoding - The Computation Problem," *Bell Syst. Tech. J.*, Vol. **45**, pp. 149–175, January 1966.

[68] C. E. Shannon, "A Mathematical Theory of Communications", *Bell Syst. Tech. J.*, **27**, pp. 379–423 (Part I), pp. 623–656 (Part II), July 1948.

[69] S. J. Simmons, "Breadth-First Trellis Decoding with Adaptive Effort," *IEEE Trans. Commun.*, **COM–38**, pp. 3-12, January 1990.

[70] G. Ungerboeck, "Channel coding with multilevel/phase signals", *IEEE Trans. Inform. Theory*, **IT–28**, pp. 55–67, January 1982.

[71] G. Ungerboeck, "Trellis Coded Modulation with Redundant Signal Sets, Part I: Introduction", *IEEE Commun. Mag.*, **25**, pp. 5–11, February 1987.

[72] G. Ungerboeck, "Trellis Coded Modulation with Redundant Signal Sets, Part II: State of the Art", *IEEE Commun. Mag.*, **25**, pp. 12–22, February 1987.

[73] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, **IT–13**, pp. 260–269, April 1967.

[74] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems", *IEEE Trans. Commun. Tech.*, **COM–19**, pp. 751–771, October 1971.

[75] A. J. Viterbi, "Wireless Digital Communication: A View Based on Three Lessons Learned", *IEEE Commun. Mag.*, **29**, pp. 33–36, September 1991.

[76] A. J. Viterbi and J. K. Omura, *Principles of Digital Communications and Coding*, McGraw-Hill, New York, 1979.

[77] A. J. Viterbi, J. K. Wolf, E. Zehavi, and R. Padovani, "A Progmatic Approach to Trellis Coded Modulation", *IEEE Commun. Mag.*, **COM–19**, pp. 11-19, July 1989.

[78] F. Q. Wang and D. J. Costello,Jr., "Erasurefree Sequential Decoding and Its Application to Trellis Codes", *Proc. 1990 IEEE Int. Symp. Inform. Theory*, p. 66, San Diego, CA, January 1990.

[79] F. Q. Wang and D. J. Costello,Jr., "A Hybrid M-algorithm/Sequential Decoder for Convolutional and Trellis Codes", *Proc. 1990 Int. Symp. Inform. Theory and Its Applications*, Hawaii, pp. 67–69, November 1990.

[80] F. Q. Wang and D. J. Costello,Jr., "Probabilistic Construction of Trellis Codes", *Proc. 1991 IEEE Int. Symp. Inform. Theory*, p. 200, Budapest, Hungary, June 1991.

[81] F. Q. Wang and D. J. Costello,Jr., "Erasurefree Sequential Decoding of Trellis Codes", submitted to the *IEEE Trans. Inform. Theory.*

[82] F. Q. Wang and D. J. Costello, Jr., "Construction of Trellis Codes for Sequential Decoding", submitted to the *IEEE Trans. Commun..*

[83] F. Q. Wang and D. J. Costello, Jr., "On the Separability of Shaping and Coding", submitted to the *IEEE Trans. Inform. Theory.*

[84] F. Q. Wang and D. J. Costello, Jr., "Construction of Robustly Good Trellis Codes", submitted to the *IEEE Trans. Inform. Theory.*

[85] F. Q. Wang and D. J. Costello, Jr., "On the Design Criteria for Trellis Codes with Sequential Decoding", *Proc. 1993 IEEE Int. Symp. Inform. Theory*, San Antonio, TX, January 1993, to appear.

[86] L. F. Wei, "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space-Part I: 180 degrees", *IEEE J. Select. Areas Commun.*, **SAC–2**, pp. 659–672, September 1984.

[87] L. F. Wei, "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space-Part II: Nonlinear Codes", *IEEE J. Select. Areas Commun.*, **SAC–2**, pp. 672–686, September 1984.

[88] L. F. Wei, "Trellis-Coded Modulation with Multi-Dimensional Constellations", *IEEE Trans. Inform. Theory*, **IT–33**, pp. 483–501, July 1987.

[89] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering,* New York, Wiley, 1965

[90] J. M. Wozencraft and R. S. Kennedy, "Modulation and Demodulation for Probabilistic Coding," *IEEE Trans. Inform. Theory*, **IT–12**, pp. 291–297, April 1966.

[91] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*, MIT Press, Cambridge, Mass., 1961.

[92] J. Wu and D. J. Costello, Jr., "New Multilevel Codes Over GF(q)," *IEEE Trans. Inform. Theory*, **IT–38**, pp. 933–939, May 1992.

[93] H. Yamamoto and K. Itoh, "Viterbi Decoding Algorithm for Convolutional Codes with Repeat Request," *IEEE Trans. Inform. Theory*, **IT–26**, pp. 540–547, September 1980.

[94] H. L. Yudkin, "Channel State Testing in Information Decoding", Sc.D. Thesis, MIT, Cambridge, Mass., 1964.

[95] E. Zehavi and J. K. Wolf, "On the Performance Evaluation of Trellis Codes," *IEEE Trans. Inform. Theory*, **IT–33**, pp. 196–202, March 1987.

[96] K. Zigangirov, "Some Sequential Decoding Procedures," *Probl. Peredachi Informatsii.*, **Vol. 2**, pp. 13–25, 1966.